
DIPLOMARBEIT

Ing.
Waitz Roman

**Additive Datenübertragung in
Barcodes von internationalen
Bahntickets**

Wien, 2014

DIPLOMARBEIT

Additive Datenübertragung in Barcodes von internationalen Bahntickets

Autor:

Ing. Waitz Roman

Studiengang:

Technische Informatik

Seminargruppe:

KI08w2wNA

Erstprüfer:

Prof. Dr.-Ing. habil. Lutz Winkler

Zweitprüfer:

Dipl. Ing. Wirth Christian

Einreichung:

Mittweida, 24.11.2014

Verteidigung/Bewertung:

Mittweida, 2014

Bibliografische Angaben:

Waitz Roman:

Additive Datenübertragung in Barcodes von internationalen Bahntickets
–2014. – X, 79, XII S.

Mittweida, Hochschule Mittweida (FH), University of Applied Sciences,
Fakultät Elektro- und Informationstechnik, Diplomarbeit, 2014

Referat:

Die vorliegende Arbeit befasst sich mit der zusätzlichen Übertragung von bahnspezifischen Daten innerhalb des, auf einem internationalen und nationalen „Home Printing“-Bahntickets aufgebrachten Barcodes. Es wird der Herstellungsprozess dieser Barcodes analysiert. Auf dieser Basis werden verschiedene Strategien zur optimalen Datenübertragung erarbeitet.

Das Hauptziel ist es, eine Strategie zu finden welche eine Datenübertragung mit dem höchst möglichen Dateninhalt realisiert.

Inhalt

Inhalt	i
Abbildungsverzeichnis	v
Tabellenverzeichnis	vii
Abkürzungsverzeichnis	x
1 Übersicht	1
1.1 Motivation	1
1.2 Ziel der Diplomarbeit	1
1.3 Kapitelübersicht	2
2 Grundlagen	3
2.1 UIC	4
2.2 RCT2-Standard /UIC918-2/	4
2.2.1 Aufbau RCT2-Fahrberechtigung	5
2.2.1.1 Feld 1 – Allgemeine Informationen	6
2.2.1.2 Feld 2 - Name und Anzahl	6
2.2.1.3 Feld 3 – Routendetails	7
2.2.1.4 Feld 4 – Klasse	7
2.2.1.5 Feld 5 – Beschreibung Zug und Sitze	7
2.2.1.6 Feld 6 – Beschreibung Tarif	8
2.2.1.7 Feld 7 – Preis	8
2.2.1.8 Feld 8 – Informationen des Betreibers	8
2.3 Internationale Online Tickets	9
2.3.1 Muster Fahrberechtigung	10
2.3.2 Graphischer Aufbau einer Fahrberechtigung	12
2.3.3 Verwaltung und Austausch von Berechtigungsnachweisen	13
2.3.4 Aufbau des Dateninhaltes des Aztec-Barcodes	14
2.3.4.1 Übersicht	14
2.3.4.2 Aufbau gesamter Dateninhalt	15
2.3.4.3 Aufbau Nutzdaten	15
2.3.4.4 U_HEAD	16
2.3.4.5 U_TLAY	17
2.3.4.6 U_TLAY Text Felder	17

2.3.4.7	Herstellerspezifische Datensätze	19
2.4	<i>Digitale Signatur</i>	20
2.5	<i>Aztec- Barcode</i>	22
2.5.1	Erstellung Aztec-Barcode.....	24
2.5.1.1	Generierung einer binären Kette	24
2.5.1.2	Ermittlung der Barcode-Parameter.....	25
2.5.1.3	Bit-Stuffing	25
2.5.1.4	Auffüllen der Codewörter	26
2.5.1.5	Anfügen der Fehlerkorrektur Bits	26
2.5.1.6	Zeichnen des Barcodes	26
2.6	<i>Daten Komprimierung</i>	27
2.6.1	LZ77	27
2.6.2	Huffman-Codierung.....	28
2.7	<i>Darstellung binärer Daten im Zeichenformat</i>	30
2.7.1	Datendarstellung im Base64-Format.....	30
2.7.2	Datendarstellung im Hexadezimalsystem	31
2.7.3	Gegenüberstellung Hexadezimal- und Base64-Darstellung	32
2.8	<i>Datendarstellung im XML-Format</i>	33
2.9	<i>Datendarstellung im JSON-Format</i>	34
2.10	<i>Datendarstellung im YAML-Format</i>	35
3	Anforderungen	36
3.1	Anforderungen durch die Vorschrift UIC-918-3	36
3.2	Konkreter Anwendungsfall Jahreskarte.....	38
3.2.1	Aufgedruckte Daten	38
3.2.2	Provider spezifische Daten	39
4	Testbeschreibung	40
4.1	<i>Testumgebung</i>	40
4.1.1	Ticket XML.....	41
4.1.2	Ticket Barcode.....	44
4.1.3	Ticket Image	44
4.1.4	Ticket HTML	45
4.2	<i>Aufbau Barcode Library</i>	45
4.3	<i>Kennzahlen</i>	47
4.4	<i>Strategien</i>	48
4.4.1	Abbildung im XML-Format	49
4.4.2	Abbildung im JSON-Format	50

4.4.3	Abbildung im YAML-Format	51
4.4.4	Abbildung als Text (SimpleText).....	52
4.4.5	Abbildung als Text komprimiert (PreCompressed)	53
4.4.6	Abbildung proprietäres Format (Complex).....	55
4.4.6.1	Darstellung Text.....	55
4.4.6.2	Darstellung Zahlen.....	57
4.4.6.3	Darstellung Datum	58
4.4.6.4	Darstellung Zeit.....	58
4.4.6.5	Bildung Bit-Kette und Base64-Text	59
4.5	Testdaten.....	60
4.5.1	Minimale Daten für Jahreskarte (Minimal)	61
4.5.2	Maximale Daten für Jahreskarte (Maximal)	62
4.5.3	Maximale Daten nach Spezifikation (MaximalSpec)	63
4.5.4	Bestehende Jahreskarte (BestehendeJK)	64
4.5.5	Unterschiedliche Daten im erweiterten Datenbereich (DifferentData)	65
5	Auswertung Ergebnisse	66
5.1	Übersicht Testergebnisse.....	67
5.2	Vergleich „SimpleText“ und „PreCompressed“	69
5.3	Vergleich Datenaustauschformate XML, JSON und YAML	70
5.4	Vergleich „SimpleText“ und „Complex“	71
5.5	Vergleich „SimpleText“ und „YAML“	72
5.6	Vergleich „BestehendeJK“ und „DifferentData“	73
6	Zusammenfassung und Ausblick	74
6.1	Zusammenfassung Testergebnisse.....	74
6.2	Eingesetzte Lösung zur Aufgabenstellung und Ausblick	75
Index		76
Literatur		77
Anlagen.....		79
Anlagen, Testergebnisse „Minimal“		I
Anlagen, Testergebnisse „Maximal“		IV
Anlagen, Testergebnisse „MaximalSpec“		VII
Anlagen, Testergebnisse „BestehendeJK“		IX

Anlagen, Testergebnisse „DifferentData“	XII
Eidesstattliche Erklärung	15

Abbildungsverzeichnis

Abbildung 1: Übersicht technische und organisatorische Grundlagen	3
Abbildung 2: Logo UIC /UIC/.....	4
Abbildung 3: Beispiel RCT2-Ticket	5
Abbildung 4: Aufbau eines RCT2-Tickets /UIC918-2/, S.81	6
Abbildung 5: Muster Fahrberechtigung.....	10
Abbildung 6: Anzeige Fahrberechtigungsmuster auf dem mobilen Gerät der ÖBB	11
Abbildung 7: Layout einer Fahrberechtigung /UIC918-3, S. 7/	12
Abbildung 8: Übersicht Dateninhalt Aztec-Barcode	14
Abbildung 9: Übersicht digitale Signatur /AnKr2012, S. 105/	20
Abbildung 10: Aztec-Barcode des Fahrtberechtigungsmusters	22
Abbildung 11: Aztec Barcode Suchmuster / HBdal2002, Seite 284/	23
Abbildung 12: Aztec Barcode Lageangabe /HBdal2002, Seite 284/	23
Abbildung 13: Aztec-Barcode Anordnung der Daten in Layern /HBdal2002, Seite 287/	26
Abbildung 14: Huffman-Baum Basis.....	28
Abbildung 15: Huffman-Baum Konstruktion der Knoten.....	28
Abbildung 16: Huffman-Baum Konstruktion Kanten.....	29
Abbildung 17: Prozess zur Erstellung eines Aztec-Barcodes nach UIC-918-3.....	37
Abbildung 18: Test Jahreskarte im RCT2-Format.....	38
Abbildung 19: Deployment Diagramm Testumgebung.....	40

Abbildung 20: UIC-918-3-konformer Aztec-Barcode als Testergebnis.....	44
Abbildung 21: Darstellung Fahrberechtigung als PNG-Image	44
Abbildung 22: Class-Diagramm Barcode-Library.....	46
Abbildung 23: Kompression-Strategie „SimpleText“ und „PreCompressed“	69
Abbildung 24: Größe komprimierte Daten – Strategie XML, JSON und YAML	70
Abbildung 25: Kompression-Strategie XML, JSON und YAML.....	70
Abbildung 26: Datengröße vor der Kompression-Strategie „SimpleText“ und „Complex“	71
Abbildung 27: Datengröße nach der Kompression-Strategie „SimpleText“ und „Complex“	71
Abbildung 28: Datengröße nach der Kompression–Strategie „SimpleText“ und „YAML“	72
Abbildung 29: Kompression-Strategie „SimpleText“ und „YAML “	72
Abbildung 30: SizeCompressed-Strategie „BestehendeJK“ und „DifferentData“.....	73
Abbildung 31: Kompression-Strategie „BestehendeJK“ und „DifferentData“	73

Tabellenverzeichnis

Tabelle 1: Beschreibung Feld 1 RCT2-Standard	6
Tabelle 2: Beschreibung Feld 2 RCT2-Standards	6
Tabelle 3: Beschreibung Feld 3 RCT2-Standards	7
Tabelle 4: Beschreibung Feld 4 RCT2-Standards	7
Tabelle 5: Beschreibung Feld 5 RCT2-Standards	8
Tabelle 6: Beschreibung Feld 6 RCT-Standards	8
Tabelle 7: Beschreibung Feld 7 RCT2-Standards	8
Tabelle 8: Beschreibung Feld 8 RCT2-Standards	8
Tabelle 9: Felder definiert durch UIC-918-3 Layout	12
Tabelle 10: Aufbau UIC-918-3 gesamter Dateninhalt des Barcode.....	15
Tabelle 11: Allgemeiner Aufbau Datensatz UIC-918-3	15
Tabelle 12: Aufbau UIC-918-3 U_HEAD Datensatz.....	16
Tabelle 13: Aufbau UIC-918-3 U_TLAY Datensatz.....	17
Tabelle 14: Aufbau UIC-918-3 U_TLAY Textfelder	17
Tabelle 15: Aufbau UIC-918-3 U_TLAY Datensatz.....	19
Tabelle 16: Beispiel Erzeugung Aztec-Barcode -> Binärkette.....	24
Tabelle 17: Aztec-Barcode Umsetzungstabelle	25
Tabelle 18: Aztec-Daten Aufteilung in Codewörter	26
Tabelle 19: Ermittelter Index aus exemplarischen Daten (LZ77)	27

Tabelle 20: Huffman-Codierung „Sessellehne“ Aufteilung Buchstaben Häufigkeit.....	28
Tabelle 21: Huffman-Codierung „Sessellehne“	29
Tabelle 22: Beispiel Ermittlung Daten im Base64-Format	30
Tabelle 23: Umsetzungstabelle Base64-Codierung.....	31
Tabelle 24: Umsetzungstabelle Hexadezimalsystem.....	31
Tabelle 25: Gegenüberstellung Darstellungsformate Hexadezimal und Base64.....	32
Tabelle 26: Daten Jahreskarte bedruckter Bereich.....	38
Tabelle 27: Daten Jahreskarte Provider spezifischer Bereich.....	39
Tabelle 28: Komponenten der Testumgebung	41
Tabelle 29: Kennzahlen für die Auswertung	47
Tabelle 30: Beispieldaten zur Darstellung der Strategien	52
Tabelle 31: Huffman-Code-Mapping Test.....	56
Tabelle 32: Parameter Texte als Huffman-Code Bitmuster.....	57
Tabelle 33: Parameter natürliche Zahlen als Bitmuster	57
Tabelle 34: Parameter Datum als Bitmuster.....	58
Tabelle 35: Parameter Uhrzeit als Bitmuster	58
Tabelle 36: Parameter „Gültig Ab“ und „Gültig Bis“ als Bitmuster	58
Tabelle 37: Testdaten des Testfalls minimale Daten für Jahreskarte.....	61
Tabelle 38: Testdaten des Testfalls maximale Daten für Jahreskarte.....	62
Tabelle 39: Testdaten des Testfalls maximale Daten nach Spezifikation.....	64
Tabelle 40: Testdaten des Testfalls bestehende Jahreskarte.....	64

Tabelle 41: Testdaten des Testfalls unterschiedliche Daten im erweiterten Datenbereich	65
Tabelle 42: Testergebnisse für die Testfälle Minimal und Maximal	67
Tabelle 43: Testergebnisse für die Testfälle MaximalSpec, BestehendeJK und DifferentData	68

Abkürzungsverzeichnis

UIC	Union internationale des chemins de fer
RCT2	Rail combined Ticket Version 2
RICS	Railway Interchange Coding System
PNR	Passenger Name Record
XML	Extensible Markup Language
W3C	World Wide Web Consortium
ÖBB	Österreichische Bundesbahn
DB	Deutsche Bahn
HTML	Hyper Text Markup Language
PNG	Portable Network Graphics
ASCII	American Standard Code for Information Interchange
YAML	YAML Ain't Markup Language
ISO	International Standard Organisation
VOR	Verkehrsverbund Ostregion
IEC	International Electrotechnical Commission

1 Übersicht

Dieses Kapitel dient zur näheren Erläuterung der Motivation und der Ziele dieser Diplomarbeit. Zusätzlich erfolgt ein Überblick über die einzelnen Kapitel dieser Arbeit.

1.1 Motivation

Beim Ausdruck von über das Internet bezogenen Bahnfahrkarten, wird zur maschinellen Überprüfung (Validierung) der Tickets ein Barcode aufgebracht, welcher dann bei der Kontrolle im Zug durch ein Validierungsgerät eingelesen wird.

Dieser Barcode enthält fahrtspezifische Daten wie den Namen des Kunden, den Preis oder auch einzelne Stationen der Reise. In der heutigen Zeit wird es immer notwendiger zusätzliche Daten zu übermitteln. Zum Beispiel Reservierungsdaten oder welche Kundenkarte einer Fahrpreismäßigung zu Grunde liegt.

Durch die Limitierungen des einzuhaltenden Standards ist die Datenkapazität, der in den Barcodes enthaltenen Daten, begrenzt.

1.2 Ziel der Diplomarbeit

Die vorliegende Arbeit befasst sich mit den einzelnen Prozessschritten des Entstehungsprozesses des Barcodes und dessen Dateninhaltes. Zusätzlich werden die dazu notwendigen technischen Grundlagen erläutert.

Das Hauptziel dieser Arbeit ist jedoch die Ermittlung einer Strategie zur Übertragung zusätzlicher Daten im Barcode.

Die Überprüfung einzelner Strategien und die Einhaltung der vorgegebenen Richtlinien werden auf Basis von einer konkreten Anforderung eines österreichischen Verkehrsverbundes mit Hilfe von Testprogrammen durchgeführt.

1.3 Kapitelübersicht

Die Diplomarbeit besteht aus 5 Kapiteln.

Nachdem im ersten Kapitel eine allgemeine Einleitung gegeben wurde, werden im Kapitel 2 die technischen und organisatorischen Grundlagen zur Erzeugung einer Fahrberechtigung erläutert.

Danach werden im 3. Kapitel die Anforderungen, welche sich aus den Vorschriften der UIC (Union internationale des chemins de fer) ergeben, sowie der konkrete Anwendungsfall zur Grundlage der Testdurchführung beschrieben.

Anschließend wird im Kapitel 4 der Aufbau und die Implementation der Testumgebung sowie die einzelnen Teststrategien erläutert.

Die Testergebnisse werden in Kapitel 5 diskutiert.

Im Kapitel 6 werden die Resultate der vorhergehenden Kapitel zusammengefasst, die Eigenleistung des Diplomanden herausgearbeitet sowie ein Ausblick auf zukünftige Lösungswege gegeben.

2 Grundlagen

Der UIC hat den Aufbau einer internationalen Fahrberechtigung für den öffentlichen internationalen Eisenbahnverkehr definiert. In diesem Kapitel wird die Organisation des UIC betrachtet. Auch die Details des Aufbaus des Barcodes einer über das Internet bezogene und auf Normalpapier selbst ausgedruckte Fahrberechtigung werden dargelegt.

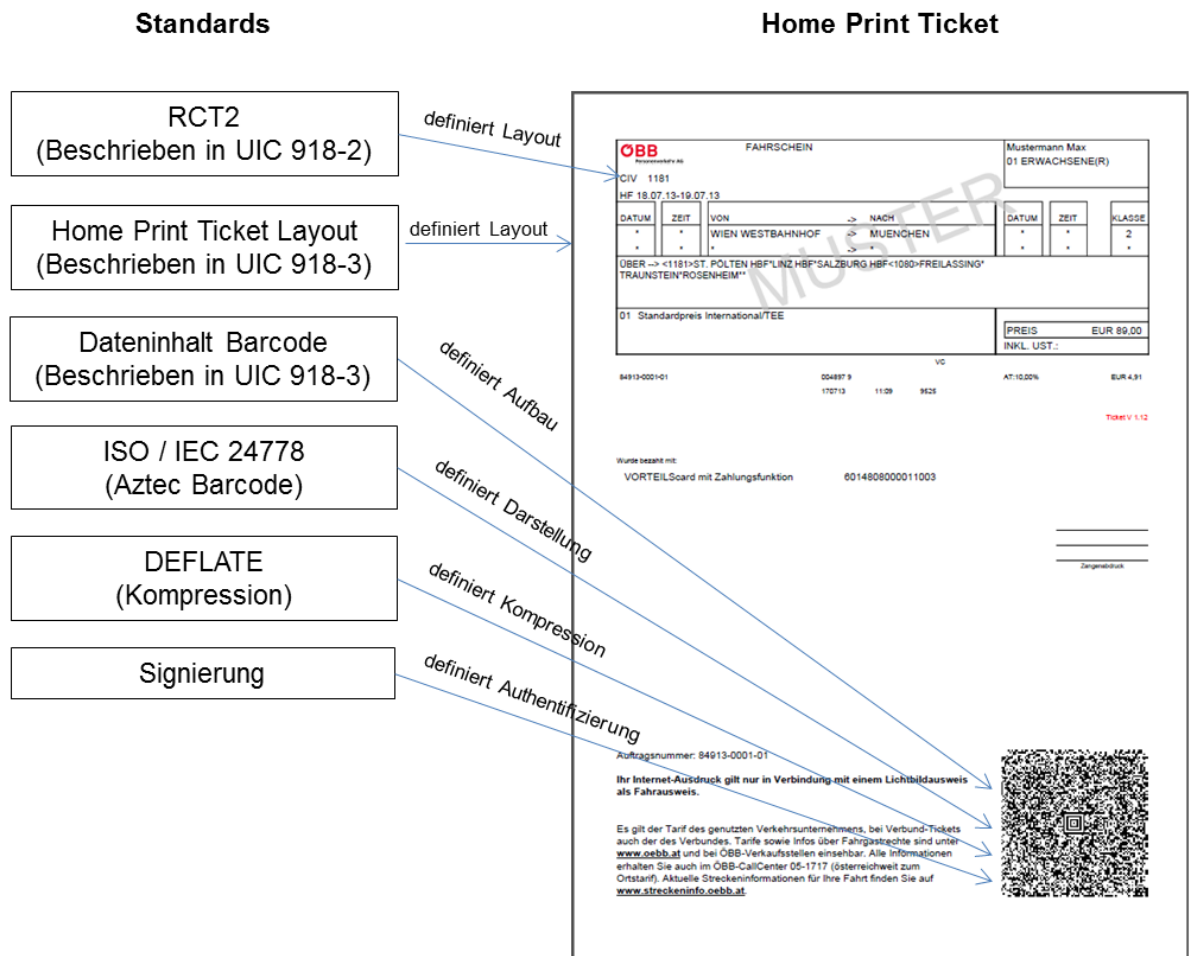


Abbildung 1: Übersicht technische und organisatorische Grundlagen

2.1 UIC



Abbildung 2: Logo UIC /UIC/

Der UIC ist ein weltweit tätiger Eisenbahnverband mit Sitz in Paris und umfasst aktuell 197 Mitglieder. Gegründet wurde der UIC am 17.10.1922 mit dem Ziel die Betriebsbedingungen der Bahnen zu vereinheitlichen.

So befasst sich der UIC mit der Vereinheitlichung aller relevanten Bereiche im grenzüberschreitenden Eisenbahnverkehr.

2.2 RCT2-Standard /UIC918-2/

Der RCT2- (Rail combined Ticket Version 2) Standard gibt den Aufbau für elektronisch ausgegebenen Transport Dokumente im internationalen Eisenbahnverkehr vor. Im nationalen Verkehr wird diese Vorschrift auch angewandt jedoch nicht immer voll erfüllt.

Beschrieben werden durch die Vorschrift UIC-918-2 die zwei folgenden Typen des Aufbaus:

- RCT2
- RCT2-compressed

Der RCT2-compressed Standard kommt nur zur Anwendung wenn der Ersteller der Fahrberechtigung zusätzliche Informationen in Form von Barcodes (Aztec- oder PDF417-Format /HBdal2002/) auf das gedruckte Ticket mitaufbringen möchte.

Für diese Arbeit ist nur der RCT2-Standard relevant daher wird der Standard RCT2-compressed nicht weiter erläutert.

Abbildung 3 zeigt beispielhafte Fahrberechtigungen, welche durch das Vertriebssystem der ÖBB (Österreichische Bundesbahn) erstellt und ausgedruckt wurden.

ÖBB Personenverkehr AG		FAHRSCHEIN		01 ERWACHSENER	
CIV 1181					
Gültig: 16.07.2013 - 15.08.2013					
DATUM	ZEIT	VON	NACH	DATUM	ZEIT
*	*	WIEN WESTBAHNHOF	-> MUENCHEN	*	*
*	*	MUENCHEN	-> WIEN WESTBAHNHOF	*	*
				KLASSE	2
ÜBER -> <1181>LINZ HBF *SALZBURG HBF<1080>FREIASSING *TRAUNSTEIN *ROSENHEIM** ***** ÜBER (-> <1080>ROSENHEIM *TRAUNSTEIN *FREIASSING *SALZBURG HBF<1181>LINZ HBF** *****					
TEE-NORMALPREIS				PREIS EUR **178,00	
AT: EUR 108,00				AT: inkl. 10% USt= EUR 9,82	
187024138 BOHRN		019342 7			
187024138		160713 14:10 409203			
Bitte ggf. Ermäßigungskarte vorweisen!					

 Personenverkehr AG		RESERVIERUNG		01 PLATZ		
CIV 1181		railjet				
DATUM	ZEIT	VON	NACH	DATUM	ZEIT	KLASSE
16.07	14:36	WIEN WEST	->MUEN HEN HBF	16.07	18:41	2
*	*	*	*	*	*	
ZUG		64 RJ WAGEN	022	STUPLATZ		
		FENSTER				
NICHTRAUCHER		GROSSRAUMWAGEN				
				PREIS EUR *****3,50		
				AT: inkl: 10% Ust= EUR 0.32		
187024139 BOHRN		432510115938				
187024139		019344 5				
		160713 14:13 409203				
		Bitte ggf. Ermüdungskarte vorzeigen!				
						

Abbildung 3: Beispiel RCT2-Ticket

Inhalte und Aufbau einer Fahrberechtigung werden durch den Standard RCT2 der Vorschrift UIC-918-2 geregelt.

Was in diesem Standard festgelegt wurde, soll nachfolgend kurz erläutert werden.

2.2.1.3 Feld 3 – Routendetails

Zeile	Spalte	Beschreibung	
5-6	02-06	Beschreibung Datum im Normalfall „DATE“.	
5-6	08-12	Beschreibung Uhrzeit im Normalfall „TIME“.	
5	14-30	„DEPARTURE“ oder „FROM“.	
5	31-34	Pfeil nach rechts im Normalfall „->“.	
5	35-51	„ARRIVAL“ oder „TO“.	
5-6	53-57	Beschreibung Datum im Normalfall „DATE“.	
5-6	59-63	Beschreibung Uhrzeit im Normalfall „TIME“.	
7	02-06	Datum der Abfahrt im Format „DD/MM“ oder „DD.MM“.	Hinfahrt
7	08-12	Zeitpunkt der Abfahrt im Format „hh.mm“.	
7	14-30	Abfahrtsstation (max. 17 Zeichen).	
7	31-34	Pfeil nach rechts im Normalfall „->“.	
7	35-51	Zielstation (max. 17 Zeichen).	
7	53-57	Datum der Ankunft im Format „DD/MM“ oder „DD.MM“.	
7	59-63	Zeitpunkt der Abfahrt im Format „hh.mm“.	Rückfahrt
8	02-06	Datum der Abfahrt im Format „DD/MM“ oder „DD.MM“.	
8	08-12	Zeitpunkt der Abfahrt im Format „hh.mm“.	
8	14-30	Abfahrtsstation (max. 17 Zeichen).	
8	31-34	Pfeil nach rechts im Normalfall „->“.	
8	35-51	Zielstation (max. 17 Zeichen).	
8	53-57	Datum der Ankunft im Format „DD/MM“ oder „DD.MM“.	
8	59-63	Zeitpunkt der Abfahrt im Format „hh.mm“.	

Tabelle 3: Beschreibung Feld 3 RCT2-Standards

2.2.1.4 Feld 4 – Klasse

Zeile	Spalte	Beschreibung	
5-6	67-71	Beschreibung Klasse im Normalfall „CLASS“.	
7	67-71	Angabe der Klasse.	Hinfahrt
8	67-71	Angabe der Klasse.	Rückfahrt

Tabelle 4: Beschreibung Feld 4 RCT2-Standards

2.2.1.5 Feld 5 – Beschreibung Zug und Sitze

Zeile	Spalte	Beschreibung
9	02-06	Text „TRAIN“, „ZUG“, ...
9	08-12	Zugnummer.
9	14-16	Zugkategorie.
9	18-25	Text „COACH“, „ZUB“, ...
9	27-29	Nummer des Zugbegleiters.

9	33-47	Text „SEAT“, „SITZ“, ...
9	49-71	Sitznummern der gebuchten Sitze.
10	33-71	Sitznummern der gebuchten Sitze.
11	02-31	Angabe ob Nichtraucher „SMOKING“ oder „NON SMOKING“.
11	33-71	Sitznummern der gebuchten Sitze.
12	02-31	Angabe ob Nichtraucher „SMOKING“ oder „NON SMOKING“.
12	33-71	Sitznummern der gebuchten Sitze.

Tabelle 5: Beschreibung Feld 5 RCT2-Standards

2.2.1.6 Feld 6 – Beschreibung Tarif

Zeile	Spalte	Beschreibung
13-15	02-03	Anzahl der Passagiere pro Tarif.
13-15	05-36	Tarifbezeichnung.
13-15	38-51	Betreiber des Tarifes.

Tabelle 6: Beschreibung Feld 6 RCT-Standards

2.2.1.7 Feld 7 – Preis

Zeile	Spalte	Beschreibung
14	53-71	Beschreibung Preis im Normalfall „PRICE“ oder „PREIS“ erweitert um die Währung.
15	53-71	Angabe des Preises

Tabelle 7: Beschreibung Feld 7 RCT2-Standards

2.2.1.8 Feld 8 – Informationen des Betreibers

Zeile	Spalte	Beschreibung
16	02-15	Buchungsnummer.
16	17-30	Beschreibung PNR (Passenger Name Record).
16	32-72	Freie Verfügung der ausstellenden Bahn.
17	02-12	Lagernummer des Papieres.
17	14-30	Eindeutige Kennung des Ausstellers.
17	32-72	Freie Verfügung der ausstellenden Bahn.
18	02-12	Nummer des ausstellenden Systems.
18	14-30	Beschreibung des Services.
18	32-36	UIC Code des Ausstellers.
18	38-45	Datum des Verkaufs im Format „DDMMYY“ oder „DD.MM.YY“ oder DD/MM/YY“.

Tabelle 8: Beschreibung Feld 8 RCT2-Standards

2.3 Internationale Online Tickets

In der Vorschrift UIC-918-3 /UIC918-3/, herausgegeben vom UIC, werden die Vorgaben für die über das Internet bezogenen und selbst auf Normalpapier ausgedruckten Fahrberechtigungen behandelt (Online Tickets).

Diese Vorschrift behandelt 3 Themen:

- Graphischer Aufbau der Fahrberechtigung (Layout)
- Verwaltung und Austausch von Berechtigungsnachweisen (Schlüsselverwaltung)
- Aufbau des Dateninhaltes des Aztec-Barcodes

2.3.1 Muster Fahrberechtigung

Zur Vereinfachung der Beschreibung der Vorgabe wurde ein Fahrberechtigungsmuster erzeugt.


ÖBB Personenverkehr AG				FAHRSCHEIN		Mustermann Max 01 ERWACHSENE(R)	
CIV 1181							
HF 18.07.13-19.07.13							
DATUM	ZEIT	VON	->	NACH	DATUM	ZEIT	KLASSE
*	*	WIEN WESTBAHNHOF	->	MUENCHEN	*	*	2
*	*	*	->	*	*	*	*
ÜBER -> <1181>ST. PÖLTEN HBF*LINZ HBF*SALZBURG HBF<1080>FREILASSING* TRAUNSTEIN*ROSENHEIM**							
01 Standardpreis International/TEE						PREIS EUR 89,00	
						INKL. UST.:	
VC							
84913-0001-01		004897 9		AT:10,00%		EUR 4,91	
		170713 11:09 9525					
Ticket V 1.12							
Wurde bezahlt mit:							
VORTEILScard mit Zahlungsfunktion				6014808000011003			
						Zangenabdruck	
Auftragsnummer: 84913-0001-01							
Ihr Internet-Ausdruck gilt nur in Verbindung mit einem Lichtbildausweis als Fahrausweis.							
Es gilt der Tarif des genutzten Verkehrsunternehmens, bei Verbund-Tickets auch der des Verbundes. Tarife sowie Infos über Fahrgastrechte sind unter www.oebb.at und bei ÖBB-Verkaufsstellen einsehbar. Alle Informationen erhalten Sie auch im ÖBB-CallCenter 05-1717 (österreichweit zum Ortstarif). Aktuelle Streckeninformationen für Ihre Fahrt finden Sie auf www.streckeninfo.oebb.at							
							

Abbildung 5: Muster Fahrberechtigung

Zur Überprüfung der Fahrberechtigung wird der Barcode des Home-Print-Tickets durch das mobile Gerät des Zugbegleiters erfasst und das Ticket wird angezeigt.

Beispielhaft die Darstellung des Fahrberechtigungsmusters auf dem mobilen Gerät der ÖBB Zugbegleiter:

Mobile10902686

File Zoom Tools Help

Ticket

Ticket V 1.4 FAHRSCHEIN

CIV 1181

HF 18.07.13-19.07.13

Mustermann Max

01 ERWACHSENE (R)

DATUM	ZEIT	VON/DE/FROM	NACH/A/TO	DATUM	ZEIT	KL.
						CL.
*	*	WIEN WESTBAHNHOF	-> MUENCHEN	*	*	2
*	*	*	-> *	*	*	*

ÜBER --> <1181>ST. PÖLTEN HBF*LINZ HBF*SALZBURG HBF<1080>FREILASSING*

TRAUNSTEIN*ROSENHEIM**

01 Standardpreis International/TEE

PREIS EUR 89,00

ESC OK

Abbildung 6: Anzeige Fahrberechtigungsmuster auf dem mobilen Gerät der ÖBB

2.3.2 Graphischer Aufbau einer Fahrberechtigung

In der Vorschrift UIC-918-3 wird als Ticketpapier ein DIN- A4- Blatt vorgeschrieben. Dieses Blatt ist in verschiedene Felder unterteilt.

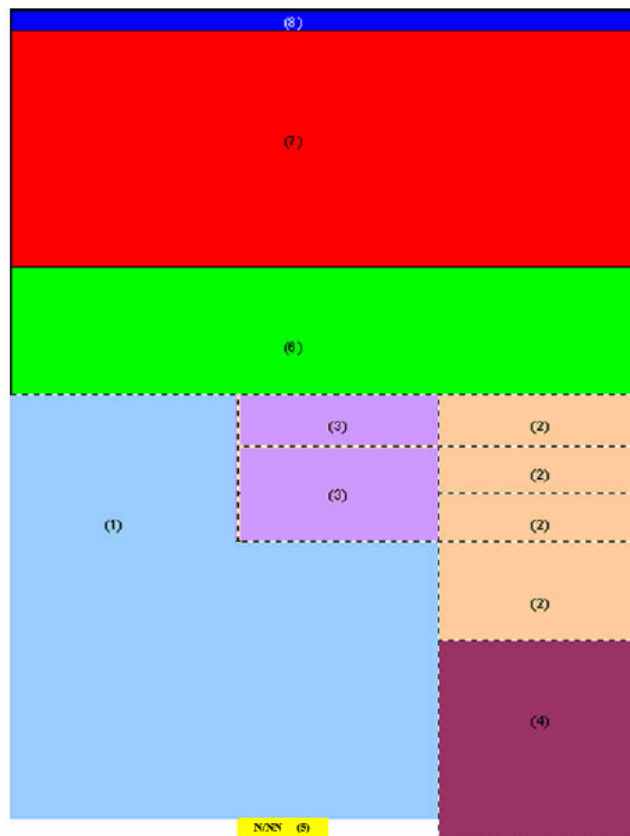


Abbildung 7: Layout einer Fahrberechtigung /UIC918-3, S. 7/

Folgende Felder sind definiert:

Feld	Farbe	Beschreibung
1	blau	Für den Austeller zur freien Verfügung.
2	orange	Zertifikatszone 1. Platz für eine eindeutige Kennung der Fahrberechtigung oder auch nur Teile davon (sollten einzelne Teilbereiche der Fahrt von verschiedenen Bahnen abgewickelt werden).
3	lila	Zertifikatszone 2 (Siehe Zertifikatszone 1 oben).
4	dunkelviolett	Zone für den 2D Barcode (Aztec-Barcode).
5	gelb	Seitennummer.
6	grün	Zone für Angaben zu möglichen Reservierungen.
7	rot	RCT2 Ticket Zone.
8	blau	Angabe des Herstellers

Tabelle 9: Felder definiert durch UIC-918-3 Layout

2.3.3 Verwaltung und Austausch von Berechtigungsnachweisen

Da bei den selbst ausgedruckten Fahrberechtigungen keinerlei Sicherheitsmerkmale auf dem Ticketpapier aufgebracht worden sind (z.B.: Wassermarken auf dem Papier, Reliefdruck, ...), ist es notwendig die ausstellende Bahn eindeutig und fälschungssicher aufzubringen. Dies geschieht mit Hilfe von elektronischer Signierung der im Barcode enthaltenen Daten. /AnKr2012, Abschnitt 6.3/

Ein Teil der Daten wird mit einem privaten Schlüssel signiert. Jene Stelle welche die Echtheit der Fahrberechtigung überprüft (normalerweise die Zugbegleiter), vollzieht die Prüfung mit Hilfe eines öffentlichen Schlüssels (siehe Abschnitt 2.4).

Ein Teil der Vorschrift UIC-918-3 beschreibt die Verwaltung der öffentlichen Schlüssel einer Bahn. Da der Vorgang für diese Arbeit nicht relevant ist, wird dieser nicht weiter erläutert.

2.3.4.2 Aufbau gesamter Dateninhalt

Im ersten Schritt ist der Dateninhalt des Barcodes wie folgt aufzuteilen. Zur Demonstration wurde der Inhalt des Barcodes des Fahrtberechtigungsmusters ermittelt.

#UT011181000030,yö>;Žh£^7\$q=UÇöäİĐöÑ—————±=\\f|öŠWR1©
 ⌘fv8ÄL00000432xœ...’İrÚ@\\fÇ_

Start	Anzahl	Name	Inhalt
0	3	Unique Message ID	#UT
3	2	Message Type Version	01
5	4	RICS Code	1181 (ÖBB)
9	5	ID signature key	00003
14	50	Signatur	
64	4	Länge komprimierter Daten	0432
86	Bis Ende	Komprimierte Daten	

Tabelle 10: Aufbau UIC-918-3 gesamter Dateninhalt des Barcode

2.3.4.3 Aufbau Nutzdaten

Ein Teil des gesamten Dateninhaltes sind die komprimierten Nutzdaten. Der Inhalt dieser Daten nach dem Dekomprimierungsvorgang gliedert sich in Datensätze (U_HEAD, U_TLAY und herstellerspezifische Datensätze) mit dem folgenden allgemeinen Aufbau:

Nummer	Zeichen	Name	Beschreibung
1	6	ID	Eindeutige Kennzeichnung des Datensatzes wie U_HEAD oder U_TLAY.
2	2	Version	Version der Datensatzstruktur
3	4	Length	Länge der enthaltenen Daten in Zeichen
4	Data	Daten des Datensatzes

Tabelle 11: Allgemeiner Aufbau Datensatz UIC-918-3

Zum Beispiel:

U_HEAD010053118184913-0001-01 1707201311091de
 U_TLAY010797RCT200400002011200012Ticket V.....
 1181AK010071Datenbereich indem eigene Daten abgespeichert werden können

Die einzelnen Datensätze werden einfach ohne Trennzeichen im Datenbereich aneinander gehängt.

2.3.4.4 U_HEAD

Zur allgemeinen Beschreibung der Fahrberechtigung dient der Datensatz U_HEAD.

Als Beispiel der U_HEAD Datensatz aus der Test –Fahrtberechtigung:

U_HEAD010053118184913-0001-01 1707201311091de..

Der Aufbau der Datensatzstruktur ist wie folgt:

Start	Anzahl	Name	Inhalt	Beschreibung
0	6	RecordID	U_HEAD	Eindeutige Bezeichnung
6	2	Record Version	01	Versionsnummer ist nach der aktuellen Version der UIC 918-3 immer 01.
8	4	Record Lenght	0053	Fixer Aufbau daher immer 0053.
12	4	RICS	1181	RICS Code des Herstellers (1181 ÖBB)
16	20	TicketId	84913-0001-01	Eindeutige Kennung der Fahrberechtigung
46	12	EditionTime	170720131109	Zeitstempel der Erzeugung mit dem Format DDM-MYYYYHHMM
58	1	Flags	1	Bit 0 = 1 .. Internationales Ticket Bit 1 = 1 .. Erzeugt bei externen Verkauf Bit 2 = 1 .. Test Ticket
59	2	Language	De	Sprache unter dessen Verwendung die Fahrberechtigung erzeugt wurde.
61	2	2nd language	-	Falls verwendet Angabe der zweiten Sprache.

Tabelle 12: Aufbau UIC-918-3 U_HEAD Datensatz

2.3.4.5 U_TLAY

Im Datensatz U_TLAY sind jene Daten enthalten, welche sichtbar im RCT2-Bereich der Fahrberechtigung aufgedruckt werden.

Daten aus dem Fahrberechtigungsmuster:

U_TLAY010797RCT200400002011200012Ticket V..

Start	Anzahl	Name	Inhalt	Beschreibung
0	6	RecordID	U_TLAY	Eindeutige Bezeichnung
6	2	Record Version	01	Versionsnummer ist nach der aktuellen Version der UIC 918-3 immer 01.
8	4	Record Lenght	0797	Länge der folgenden Daten.
12	4	Layout	RCT2	Bezeichnung des Layouts.
16	4	Fieldcount	0040	Anzahl der folgenden Text Felder.
20	..	Data	..	Dateninhalt des U_TLAY Datensatzes.

Tabelle 13: Aufbau UIC-918-3 U_TLAY Datensatz

2.3.4.6 U_TLAY Text Felder

Der Dateninhalt des U_TLAY Datensatzes gliedert sich auf in einzelne Textfelder die mit Hilfe des folgenden Formates beschrieben werden.

Daten aus dem Fahrberechtigungsmuster:

0053011400014Mustermann Max

Start	Anzahl	Name	Inhalt	Beschreibung
0	2	Field Line	00	Zeile
2	2	Field column	53	Spalte
4	2	Field height	01	Höhe Textfeld
6	2	Field width	14	Länge Textfeld
8	1	Field format	0	Schriftformat des Textes (normal, fett, kursiv, klein und Kombinationen)
9	4	Text length	0014	Länge des Textes in Zeichen
13	end	Text	Mustermann Max	Inhalt des Textes

Tabelle 14: Aufbau UIC-918-3 U_TLAY Textfelder

Anhand des Datensatzes sieht man, dass in Zeile 0 an der Stelle 53 der Text „Mustermann Max“ mit der Länge von 14 Zeichen steht. Die Textfelder des Fahrberechtigungs-musters sind:

00400002011200012Ticket V 1.4
0019011000010FAHRSCHEIN
0053011400014Mustermann Max
015301160001601 ERWACHSENE(R)
0202010300003CIV
02060104000041181
0211010000000
0253010000000
0302012000020HF 18.07.13-19.07.13
0352010000000
0603010100001*
0609010100001*
0614011600016WIEN WESTBAHNHOF
0631010200002->
0635010800008MUENCHEN
0654010100001*
0660010100001*
06670101000012
0703010100001*
0709010100001*
0714010100001*
0731010200002->
0735010100001*
0754010100001*
0760010100001*
0767010100001*
0802017100071ÜBER --> <1181>ST. PÖLTEN HBF*LINZ HBF*SALZBURG
HBF<1080>FREILASSING*
0902012200022TRAUNSTEIN*ROSENHEIM**
1002010000000
1102010000000
120201020000201
1205013100031Standardpreis International/TEE
1353010500005PREIS
1359010900009EUR 89,00
1402010000000
1438010000000

2.3.4.7 Herstellerspezifische Datensätze

Die Vorschrift UIC-918-3 erlaubt es dem Hersteller der Fahrberechtigung, spezifische Daten zusätzlich im Datenbereich des Barcodes abzulegen. Dafür ist ein eigener Datensatz mit dem folgenden allgemeinen Aufbau definiert:

Ein willkürliches Beispiel:

1181AK010071Datenbereich indem eigene Daten abgespeichert werden können

Start	Anzahl	Name	Inhalt	Beschreibung
0	6	RecordID	1181AK	Eindeutige Bezeichnung
6	2	Record Version	01	Versionsnummer des Datensatzes.
8	4	Record Length	0071	Länge des Datensatzes.
12	..	Daten	Datenbereich	Spezifischer Dateninhalt des Datensatzes.

Tabelle 15: Aufbau UIC-918-3 U_TLAY Datensatz

Der eindeutige Bezeichner des Datensatzes setzt sich zusammen aus der RICS-Nummer des Herstellers (aus dem Beispiel 1181) und einer freien Kennung bestehend aus zwei Zeichen (aus dem Beispiel AK).

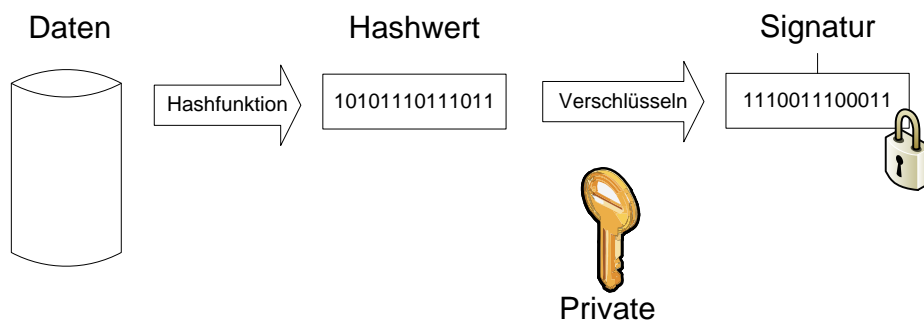
Genau hier sieht die UIC die Möglichkeit vor, zusätzliche Daten in den Barcode zu integrieren. Die Schwierigkeit ist es jedoch möglichst viele Informationen entsprechend den Vorgaben einzubetten. Dies wird hier im weiteren Verlauf durch den Diplomanden erarbeitet.

2.4 Digitale Signatur

Wenn Fahrtberechtigungen über das Internet bezogen und direkt vom Kunden selbst ausgedruckt werden, ist es besonders wichtig mit Sicherheit den Aussteller der Fahrtberechtigung eindeutig zu identifizieren.

Dies geschieht mit Hilfe der digitalen Signatur.

Signierung



Überprüfung

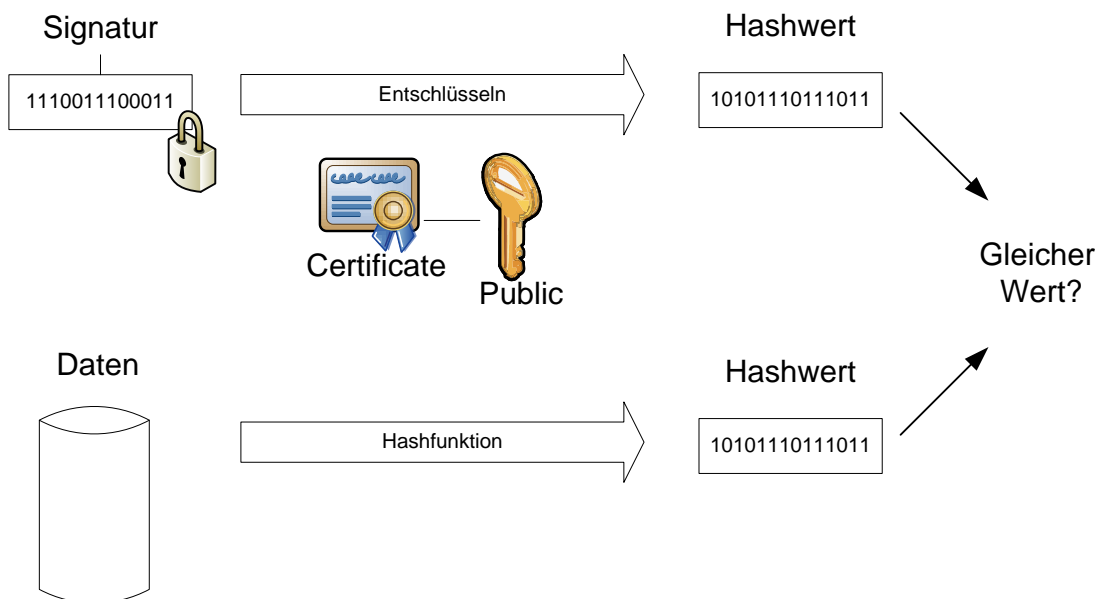


Abbildung 9: Übersicht digitale Signatur /AnKr2012, S. 105/

„The purpose of a digital signature is to provide a means for an entity to bind its identity to a piece of information.“ /HBaC1996, Abschnitt 1.6/

Die digitale Signatur ist ein kryptographisches Verfahren zur Bindung von einer Identität und Informationen (Daten). Dazu wird mit Hilfe einer Hash-Funktion ein Wert auf Basis der Daten errechnet und mit einem privaten Schlüssel verschlüsselt. Ein öffentlicher Schlüssel (passend zum privaten Schlüssel) und die Daten ermöglichen es nun jedem zu überprüfen ob die errechnete Signatur vom Besitzer des privaten Schlüssels erstellt worden ist.

„A hash function is a computationally efficient function mapping binary strings of arbitrary length to binary strings of some fixed length, called hash-values.“

/HBaC1996, Abschnitt 1.9/

Im Falle der durch die Vorschrift UIC-918-3 beschriebenen Darstellung der Fahrberechtigung wird als der zu signierende Datenbereich der komplette komprimierte Datenteil herangezogen.

Vom Hersteller wird ein Zertifikat (beinhaltet den öffentlichen Schlüssel) zur Überprüfung des Herstellers zur Verfügung gestellt.

2.5 Aztec- Barcode

Ein Barcode ist eine optisch und elektronisch lesbare Darstellung von Daten, bestehend aus verschiedenen breiten Strichen, Punkten und dazwischen liegenden Lücken. Es gibt Barcodes in einer eindimensionalen (zum Beispiel Strichcodes auf Waren im Supermarkt) und einer zweidimensionalen Ausprägung unter anderem dem Aztec-Barcode.



Abbildung 10: Aztec-Barcode des Fahrtberechtigungsmusters

Der Aztec-Barcode ist genormt nach ISO/IEC 24778 (International Standard Organisation/ International Electrotechnical Commission) und besteht aus einem in der Mitte liegenden Suchelement (zusammengesetzt aus ineinander verschachtelten Quadraten) sowie örtlich um das Suchelement herum angeordnete Datenfelder (ebenfalls quadratisch).

„Der Aztec Code in seiner Standardform erlaubt die flexible Codierung großer Datenmengen auf kleiner Fläche.“ /HBdal2002, Abschnitt 4.8.4.1/

„Sehr schnell auffindbar durch das prägnante Suchmuster. Kostengünstig herzustellen, da alle Standard Drucktechniken verwendbar sind.“ /HBdal2002, Abschnitt 4.8.4.3/

Ein Vorteil des Aztec-Barcodes ist es, dass man bei teilweiser Zerstörung des Barcodes (Druck bleicht aus, Unterlage zerknittert, ...) trotzdem noch den ganzen Inhalt lesen kann. Es ist hier möglich eine Datensicherheit von bis zu 95% (bis zu 95% nicht lesbar), bezogen auf das Datenfeld des Barcodes abzubilden. Das Verfahren zur Bildung der für die Datensicherheit zuständigen Bereiche ist das Reed-Solomon-Verfahren. Eine höhere Datensicherheit zieht eine Verringerung der Nutzdatenmenge bei gleichbleibendem Platz mit sich. /UrlReedS/

Nach UIC-918-3 wird eine Datensicherheit von 23% verlangt. /UIC918-3, S. 60/

Die Daten ordnen sich spiralförmig um ein zentrales Suchmuster bestehend aus mehreren schwarzen Quadraten. Beim Einlesen des Barcodes wird zuerst nach dem Suchmuster gesucht und dann wird der Aztec-Barcode von innen nach außen analysiert. Dem entsprechend benötigt der Aztec-Barcode keine Ruhezeiten (helle Flächen rund um den Barcode) zur Abgrenzung.

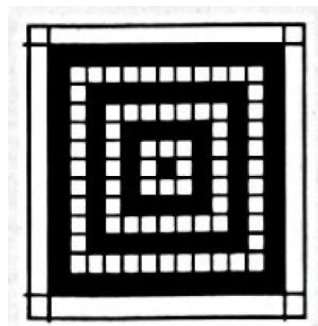


Abbildung 11: Aztec Barcode Suchmuster / HBdal2002, Seite 284/

Nach dem das Suchmuster des Aztec-Barcodes beim Einlesen gefunden wurde, wird die Lage (Rotation) anhand der Lageangabe am äußeren Rand des Suchmusters ermittelt. „An jeder Ecke des Codes befindet sich ein winkelförmiges Codemuster bestehend aus drei Zellen. Die linke untere Ecke ist durch drei weiße Zellen gekennzeichnet. Rechts unten ist eine Zelle schwarz. Rechts oben sind zwei Zellen schwarz und links oben ist der Lagewinkel mit drei schwarzen Zellen zu finden“ / HBdal2002, Seite 284/

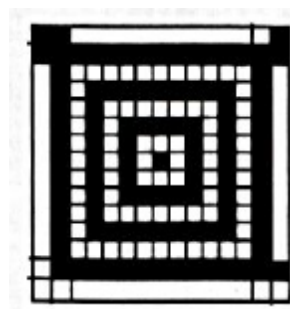


Abbildung 12: Aztec Barcode Lageangabe /HBdal2002, Seite 284/

2.5.1 Erstellung Aztec-Barcode

Die Generierung des Aztec-Barcodes aus den Daten erfolgt in mehreren Schritten:

1. Generierung einer binären Kette
2. Ermittlung der Barcode-Parameter
3. Bit-Stuffing (siehe Abschnitt 2.5.1.3)
4. Auffüllen der Code Wörter
5. Anfügen der Fehlerkorrektur-Bits
6. Zeichnen des Barcodes

2.5.1.1 Generierung einer binären Kette

Anhand des folgenden Beispiels soll verdeutlicht werden wie der Text „Test Daten“ in eine Kette aus Bits anhand der weiter unten angegebenen Tabelle in einen Aztec-Zeichencode umgewandelt wird.

Daten	T	L/L	E	s	t	D/L	SP	U/L
Aztec dec	21	28	6	20	21	30	1	29
Binär	10101	11100	00110	10100	10101	11110	00001	11101
Daten	D	L/L	a	t	e	n	P/S	!
Aztec dec	5	28	2	21	6	15	0	6
Binär	00101	11100	00010	10101	00110	01111	00000	00110

Tabelle 16: Beispiel Erzeugung Aztec-Barcode -> Binärkette

Das ergibt folgende Kette von 80 Bits:

10101111000011010100101011111000001111010010111100000101010100110011110
000000110

Code	Mode					Code	Mode				
	Upper	Lower	Mixed	Punct	Digit		Upper	Lower	Mixed	Punct	Digit
0	P/S	P/S	P/S	FLG(<i>n</i>)	P/S	16	O	o	^\	+	
1	SP	SP	SP	CR	SP	17	P	p	^]	,	
2	A	a	^A	CR LF	0	18	Q	q	^^	-	
3	B	b	^B	. SP	1	19	R	r	^_	.	
4	C	c	^C	, SP	2	20	S	s	@	/	
5	D	d	^D	: SP	3	21	T	t	\	:	
6	E	e	^E	!	4	22	U	u	^	;	
7	F	f	^F	"	5	23	V	v	_	<	
8	G	g	^G	#	6	24	W	w	`	=	
9	H	h	^H	\$	7	25	X	x		>	
10	I	i	^I	%	8	26	Y	y	~	?	
11	J	j	^J	&	9	27	Z	z	^?	[
12	K	k	^K	'	,	28	L/L	U/S	L/L]	
13	L	l	^L	(.	29	M/L	M/L	U/L	{	
14	M	m	^M)	U/L	30	D/L	D/L	P/L	}	
15	N	n	^[*	U/S	31	B/S	B/S	B/S	U/L	

Tabelle 17: Aztec-Barcode Umsetzungstabelle

2.5.1.2 Ermittlung der Barcode-Parameter

Anhand der Länge der binären Daten wird nun die Größe des Barcodes und der einzelnen Code-Worte ermittelt.

In der Vorschrift UIC-918-3 wird eine Größe des Aztec-Barcodes von 17 Layern gefordert, das entspricht einer Größe von 87*87 Blöcken. /UIC918-3, Abschnitt 8.7/

Die Größe der Codewörter ist somit auf 10 Bits und der maximale Dateninhalt mit 621 Bytes festgelegt. /HBdal2002, Abschnitt 4.8.4.7/

2.5.1.3 Bit-Stuffing

In diesem Schritt wird die Binärkette in die zuvor ermittelte Größe eines Codewortes geteilt. Um ganze Codewörter komplett gefüllt mit 0 oder 1 zu vermeiden werden diese Codewörter speziell gekennzeichnet.

Binärkette:

1010111100001101010010101111000001111010010111100000101010100110011110
000000110

Die Codewörter sind in der folgenden Tabelle beschrieben:

Codewort	Inhalt
0	1010111100
1	0011010100
2	1010111110
3	0000111101
4	0010111100
5	0001010101
6	0011001111
7	0000000110

Tabelle 18: Aztec-Daten Aufteilung in Codewörter

2.5.1.4 Auffüllen der Codewörter

Sollte anhand der Binärkette das letzte Codewort nicht die errechnete Größe haben wird dieses entsprechend aufgefüllt. Zusätzlich wird nun überprüft, ob der ganze Datenbereich des Barcodes mit den Daten ausgenutzt wurde. Sollte dies nicht der Fall sein wird auch hier der restliche Datenbestand mit Leerdaten aufgefüllt.

2.5.1.5 Anfügen der Fehlerkorrektur Bits

Es wird nun die Fehlerkorrektur nach Reed-Solomon errechnet und den entsprechenden Datenbereichen hinzugefügt. /UrlReedS/

2.5.1.6 Zeichnen des Barcodes

Im letzten Schritt werden die ermittelten Daten (Datenbereich inklusive Fehlerkorrektur) in Layern spiralförmig rund um den Kern im Uhrzeigersinn gezeichnet.

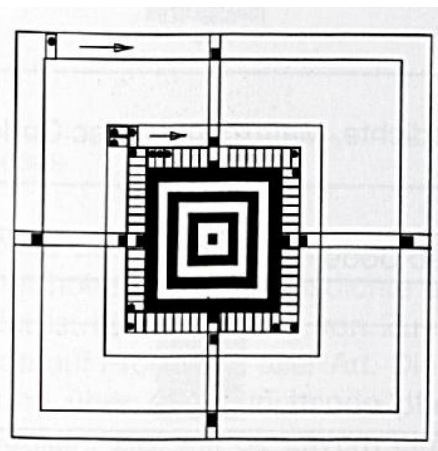


Abbildung 13: Aztec-Barcode Anordnung der Daten in Layern /HBdal2002, Seite 287/

2.6 Daten Komprimierung

Wie in „Aufbau des Dateninhaltes des Aztec-Barcodes“ beschrieben wird ein Teil des Dateninhalts des Aztec-Barcodes mittels eines Datenkomprimierungsverfahrens verkleinert.

Die Datenkomprimierung ist ein Verfahren zur Verkleinerung der Datenmenge. In diesem Fall wird der verlustfreie DEFLATE-Algorithmus /InDaCo2005, Abschnitt 5.5/ angewandt.

In einem DEFLATE-Komprimierungsvorgang werden zuerst nach dem LZ77-Algorithmus mehrfach vorhandene Daten reduziert und danach eine Huffman-Codierung abgearbeitet.

2.6.1 LZ77

Das LZ77-Verfahren basiert auf dem 1977 von Abraham Lempel und Jacob Ziv veröffentlichten LZ1-Verfahren. /InDaCo2005, Abschnitt 5.4.1/

In diesem Verfahren wird der ganze Datenbereich auf mehrfach vorkommende gleiche Datenblöcke durchsucht. Sollten Datenblöcke mehrfach vorkommen, werden diese in einen Index aufgenommen und im Datenbereich durch eine Referenz ersetzt.

Als Beispiel verwenden wir die folgenden Ausgangsdaten:

ABCD1234567890CDEFG1234567890HIJK234567890BCD

Ermittelter Index:

REF	Inhalt
0	BCD
1	1234567890
2	234567890

Tabelle 19: Ermittelter Index aus exemplarischen Daten (LZ77)

Ermittelte Daten (exklusive Index):

A[REF 0][REF 1]CDEFG[REF 1]HIJK[REF 2][REF 0]

In Folge können dann mit Hilfe des Index und den Referenzen die vollständigen Daten verlustfrei wieder hergestellt werden.

2.6.2 Huffman-Codierung

Bei der, im Jahr 1952 von David A. Huffman entwickelten Codierung, wird jedem Zeichen ein Code zugewiesen. Je öfter ein Zeichen verwendet wird, desto kleiner soll der für das Zeichen stehende Code sein. /InDaCo2005, S. 41/

Als Beispiel wird nun das Wort „SESSELLEHNE“ nach der Huffman-Codierung komprimiert:

Das Wort besteht aus den Buchstaben:

Buchstabe	E	S	L	H	N
Häufigkeit	4	3	2	1	1

Tabelle 20: Huffman-Codierung „Sessellehne“ Aufteilung Buchstaben Häufigkeit

Im Anschluss wird nun der Huffman-Baum in mehreren Schritten gebildet:

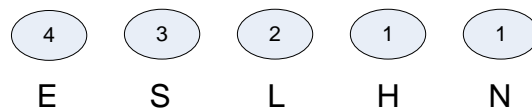


Abbildung 14: Huffman-Baum Basis

Es werden nun die Knoten mit den kleinsten Häufigkeiten zu einem neuen Knoten zusammengefasst. Der neue Knoten wird mit der Summe der Häufigkeiten markiert. Dieser Vorgang wird so lange wiederholt, bis alle Zeichen in einem Baum untergebracht worden sind.

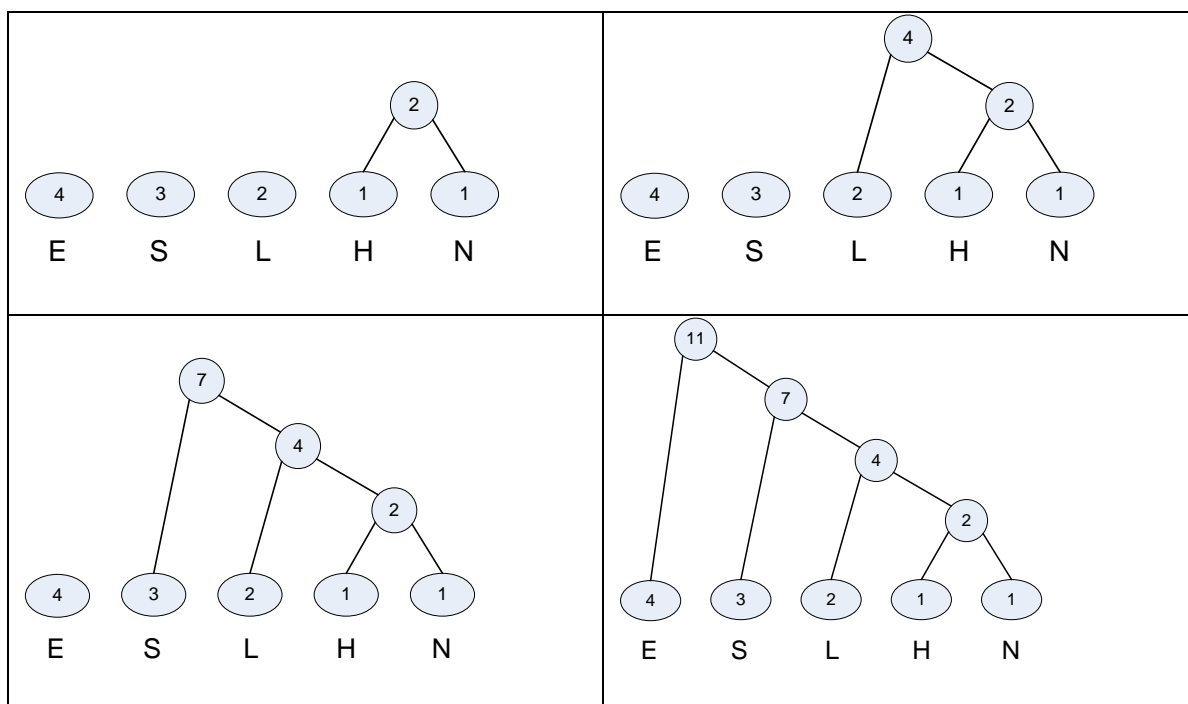


Abbildung 15: Huffman-Baum Konstruktion der Knoten

Nach dem der komplette Baum gebildet wurde, werden nun die Kanten (links mit 1 und rechts mit 0) beschriftet.

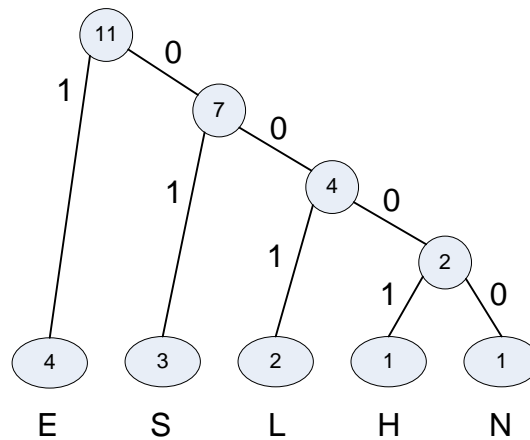


Abbildung 16: Huffman-Baum Konstruktion Kanten

Daraus resultiert die folgende Codierung:

Buchstabe	Codierung
E	1
S	01
L	001
H	0001
N	0000

Tabelle 21: Huffman-Codierung „Sessellehne“

Das Wort „SESSELLEHNE“ kann nun wie folgt in 24 Bit verlustfrei codiert werden:

S	E	S	S	E	L	L	E	H	N	E
01	1	01	01	1	001	001	1	0001	0000	1

2.7 Darstellung binärer Daten im Zeichenformat

Bei einzelnen Strategien zur Unterbringung zusätzlicher Informationen im UIC-918-3-konformen Aztec-Barcode ist es notwendig, binäre Daten abzubilden. Da es aber nur möglich ist, Zeichen aus dem ASCII-Zeichensatz /UrlASCII/ (American Standard Code for Information Interchange) zu verwenden, muss man die binären Daten in ein kompatibles Format umwandeln.

Zur Auswahl stehen 2 Formate, die im Folgenden beschrieben und im Anschluss gegenüber gestellt werden:

2.7.1 Datendarstellung im Base64-Format

Für die Übermittlung von binären Daten in Form von Zeichen gibt es mehrere Möglichkeiten. Eine dieser Möglichkeiten ist das Base64-Format /UrlBase64/. Dabei werden binäre Daten als eine Zeichenkette von ASCII-Zeichen /UrlASCII/ dargestellt. Sehr verbreitet ist dieses Format zur Übermittlung von Anhängen in e-mails.

Mit Hilfe dieser Methode werden jeweils drei Bytes mit jeweils acht Bit (drei mal acht Bit ergibt eine Länge von 24 Bit) in vier Base64-kodierte Zeichen (jeweils sechs Bit) umgesetzt, welche nach der folgenden Tabelle (siehe Tabelle 23: Umsetzungstabelle Base64-Codierung) einem Buchstaben entsprechen.

Als Beispiel werden aus den 3 Bytes (127, 0, 42) in Base64-Format „fwAq“.

Byte 1								Byte 2								Byte 3							
127								0								42							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0
Zeichen 1						Zeichen 2						Zeichen 3						Zeichen 4					
5	4	3	2	1	0	5	4	3	2	1	0	5	4	3	2	1	0	5	4	3	2	1	0
0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0
31						48						0						42					
F						w						A						Q					

Tabelle 22: Beispiel Ermittlung Daten im Base64-Format

Bei der Umsetzung benötigt die Base64-Darstellung 33% mehr Datenvolumen.

Die Umsetzung der übersetzten binären Daten in Zeichen erfolgt nach der folgenden Tabelle:

Wert	Zeichen	Wert	Zeichen	Wert	Zeichen	Wert	Zeichen
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/

Tabelle 23: Umsetzungstabelle Base64-Codierung

2.7.2 Datendarstellung im Hexadezimalsystem

Zahlen (Daten) werden im Hexadezimalsystem zur Basis 16 dargestellt. Üblicherweise werden jeweils 8 Bit abgebildet und entsprechend dargestellt.

Hexadezimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Dezimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Tabelle 24: Umsetzungstabelle Hexadezimalsystem

Zum Beispiel:

Dezimal 135 = $8 \cdot 16^1 + 7 \cdot 16^0$ = Hexadezimal 87

Dezimal 232 = $14 \cdot 16^1 + 8 \cdot 16^0$ = Hexadezimal E8

Bei dieser Umwandlung verdoppelt sich das Datenaufkommen.

2.7.3 Gegenüberstellung Hexadezimal- und Base64-Darstellung

Das Ziel ist die möglichst geringe Größe der Daten nach einer Kompression mit DEFLATE zu erreichen. Um das beste Format zu ermitteln wurden verschiedene binäre Daten in das entsprechende Format umgewandelt und danach komprimiert.

Die jeweils grün hinterlegten Felder bezeichnen die bessere Komprimierung basieren auf der Größe der Ausgangsdaten.

Ausgangsdaten [Byte]	ASCII-Darstellung		DEFLATE komprimiert			
	Hex [Byte]	Base64 [Byte]	Hex [Byte]	Hex [%]	Base64 [Byte]	Base64 [%]
226	452	304	241	106,64	256	113,27
507	1014	676	502	99,01	506	99,80
926	1852	1236	1023	110,48	960	103,67
1462	2924	1952	1641	112,24	1499	102,53
2610	5220	3480	2978	114,10	2652	101,61
3699	7398	4932	4239	114,60	3752	101,43

Tabelle 25: Gegenüberstellung Darstellungsformate Hexadezimal und Base64

Es ergeben sich nach der Kompression mit DEFLATE leichte Vorteile für die Darstellung im BASE64-Format.

2.8 Datendarstellung im XML-Format

XML (Extensible Markup Language) dient zur Darstellung von strukturierten Daten im TEXT-Format. Erstmals spezifiziert durch das W3C (World Wide Web Consortium) im Februar 1998, hat sich XML als Standard für den plattform- und implementationsunabhängigen Datenaustausch zwischen mehreren Systemen etabliert. Zum Beispiel ist XML die Basis für Webservice–Aufrufe im Internet. Es dient mit SyncML unter anderem auch als Basis zur Synchronisation von Kalenderdaten.

Eine Darstellung von Daten basierend auf XML besteht aus mehreren strukturell angeordneten Entitäten. Jede Entität besteht aus einem Datenbereich welcher durch einen Start-Tag eingeleitet und einem End-Tag beendet wird.

Als Beispiel:

```
<Familienvater >  
  <Name>„Max Mustermann</Name>  
  <Alter> 42</Alter>  
  <Verheiratet>ja</ Verheiratet>  
  <Kinder>  
    <Name>Maximilian</Name>  
    <Name>Susanne</Name>  
    <Name>Erwin</Name>  
  </Kinder>  
</Familienvater >
```

Das Beispiel beschreibt einen 42 jährigen Familienvater mit dem Name Max Mustermann. Er ist verheiratet und hat drei Kinder (Maximilian, Susanne und Erwin).

2.9 Datendarstellung im JSON-Format

Eine weitere auf Zeichen basierende Darstellungsform ist JSON.

„JSON (JavaScript Object Notation) ist ein schlankes Datenaustauschformat, das für Menschen einfach zu lesen und zu schreiben und für Maschinen einfach zu parsen (Analysieren von Datenstrukturen) und zu generieren ist.“ /UrlJSON/

JSON basiert auf einer Untermenge der Programmiersprache „javascript“ und ist sehr gebräuchlich im Internet zum Austausch von Daten. Aufgrund der weiten Verbreitung gibt es für alle gängigen Programmiersprachen Implementierungen zum Erstellen und Lesen von Daten im JSON-Format.

Ein JSON-Objekt besteht aus einem Namen, ein oder mehreren Name/Wert Paaren und / oder ein oder mehreren Listen von Werten (Array).

Zum Beispiel:

Familienvater

```
{
    „Name“ : „Max Mustermann“,
    „Alter“ : 42,
    „Verheiratet“ : true,
    „Kinder“ : [ „Maximilian“, „Susanne“, „Erwin“ ];
}
```

Wie das vorige Beispiel beschreibt dieses einen 42 jährigen Familienvater mit dem Name Max Mustermann. Er ist verheiratet und hat drei Kinder (Maximilian, Susanne und Erwin).

2.10 Datendarstellung im YAML-Format

YAML (YAML Ain't Markup Language) ist ähnlich JSON ein schlank gehaltenes und benutzerfreundliches Datenaustauschformat. Ein Vorteil gegenüber anderen Formaten ist, dass weniger Zeichen und somit weniger Datenvolumen für die Darstellung der Datenstruktur verwendet werden. /UrlYAML/

Ein YAML-Objekt besteht wie ein JSON-Objekt aus ein oder mehreren Name/Wert Paaren und / oder ein oder mehreren Listen von Werten (Array).

Zum Beispiel:

```
{ Name: Max Mustermann, Alter: '42', Verheiratet: Ja, Kinder: [Maximilian, Susanne, Erwin]}
```

Der Inhalt der Daten ist der gleiche wie bei den Beispielen davor.

3 Anforderungen

In diesem Kapitel werden die sich aus der Vorschrift UIC-918-3 ergebenden Anforderungen zusammengefasst und daraus ein Prozess zur Erstellung des Aztec-Barcodes durch den Diplomanden ermittelt.

Des Weiteren wird in diesem Kapitel die konkrete Anforderung eines österreichischen Verkehrsverbundes dargestellt, welche als Basis für den Testvorgang dient.

3.1 Anforderungen durch die Vorschrift UIC-918-3

Aufgrund der im Abschnitts 2.3.4 beschriebenen Vorschrift UIC-918-3, ergeben sich die folgenden Einschränkungen:

- Der Dateninhalt des Barcodes darf nicht größer sein als 621 Bytes.
- Der ermittelte Barcode darf nicht mehr als 17 Layer (87x87) beinhalten. Dies entspricht einem maximalen Dateninhalt von 621 Bytes bei einer Fehlerkorrektur von 23%.

Sollte sich der Aztec-Barcode den Anforderungen entsprechend nicht erzeugen lassen, so sind nach UIC-918-3 die „Provider spezifischen Daten“ zu reduzieren.

Aus diesen Anforderungen ergibt sich folgender Prozess zur Erstellung einer Fahrberechtigung in Form eines Aztec-Barcodes nach UIC-918-3.

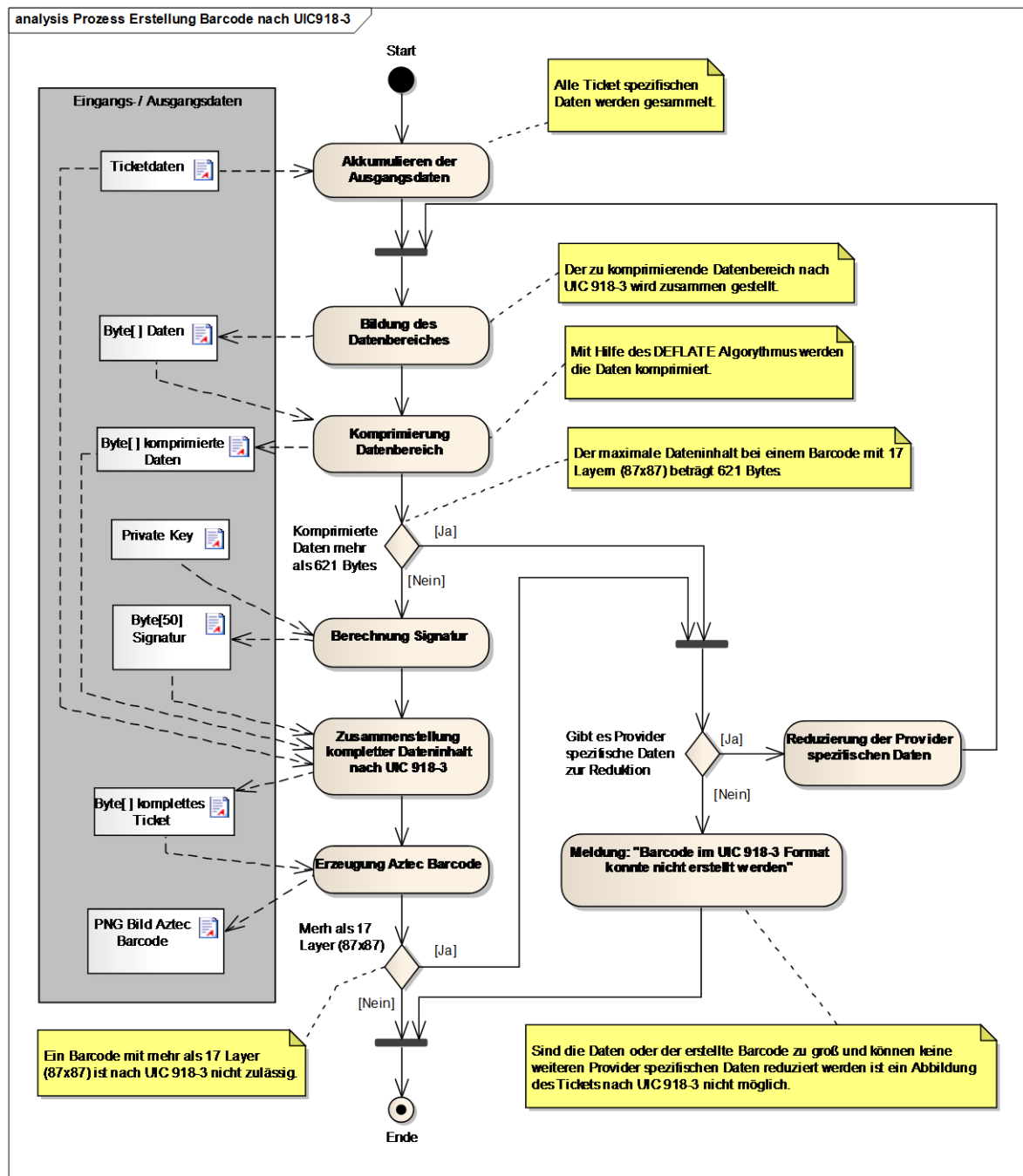


Abbildung 17: Prozess zur Erstellung eines Aztec-Barcodes nach UIC-918-3

3.2 Konkreter Anwendungsfall Jahreskarte

Als konkrete Anforderung wird die Jahreskarte, ausgestellt von einem österreichischen Verkehrsverbund, herangezogen. Auf dieser wird zukünftig ein Barcode nach UIC-918-3 aufgebracht werden.

Die Test-Fahrberechtigung (Jahreskarte) in der RCT2-Darstellung:

Fahrschein				Waitz Roman Personen 01	
CIV Gültig von 01.01.2013 00:01 bis 31.12.2013 23:59					
Datum	Zeit	Von -> Nach		Datum	Zeit
		Wien Hütteldorf Purkersdorf / Gablitz			Klass.
Über Purkersdorf					
Gilt nur in Verbindung mit einem gültigen Lichtbildausweis					
				EUR 034,20.-	

Abbildung 18: Test Jahreskarte im RCT2-Format

3.2.1 Aufgedruckte Daten

Folgende Daten befinden sich im gedruckten Bereich der Jahreskarte.

Name	Länge	Beschreibung
Vorname	10	Vorname des Fahrgastes
Familienname	10	Familienname des Fahrgastes
Von	20	Beginn des örtlichen Gültigkeitsbereiches der Jahreskarte
Nach	20	Ende des örtlichen Gültigkeitsbereiches der Jahreskarte
Gültig Ab	16	Start-Zeitpunkt der Gültigkeit im Format ,DD.MM.YYYY HH:MM'
Gültig Bis	16	End-Zeitpunkt der Gültigkeit im Format ,DD.MM.YYYY HH:MM'
Preis	8	Preis der Jahreskarte im Format „xxx.xx.“
Zusätzlicher Text	200	Zusätzlicher beschreibender Text (z.B.: Via, Einschränkungen, Berechtigungsnachweise, ...)
Anzahl der Personen	1	Anzahl der reisenden Personen (bei der Jahreskarte immer 1)

Tabelle 26: Daten Jahreskarte bedruckter Bereich

3.2.2 Provider spezifische Daten

Zusätzlich, zur besseren maschinellen Überprüfung ist es nach Vorgabe des Verkehrsverbundes notwendig, weitere Daten im Bereich der Provider spezifischen Daten unter zu bringen.

Diese Daten sind wie folgt:

Name	Länge	Beschreibung												
Währungskennzeichen	1	‚E‘ für Euro												
KundenId	18	Eindeutige Kennung des Kunden												
PartnerId	10	Eindeutige Kennung des ausstellenden Partner-Betriebs												
EfsFGGeschlecht	1	M Männlich W Weiblich												
EfsFGGeb	8	Geburtstag des Fahrgastes im Format ‚ddmmyyy‘												
EfsFGName	25	Namen des Fahrgastes												
EfsIDTyp	2	Ausweis Typ												
EfsID	10	Eindeutige Kennung des Ausweises												
Artikel	6/Artikel	Mehrere Artikel möglich. Ein Artikel wird beschrieben durch: <table border="1"> <tr> <th>Name</th><th>Länge</th><th>Beschreibung</th></tr> <tr> <td>ID</td><td>3</td><td>Eindeutige Kennung des Artikels</td></tr> <tr> <td>Kennung</td><td>1</td><td>B ... Basisartikel S ... Subartikel</td></tr> <tr> <td>Anzahl</td><td>2</td><td>Anzahl des Artikels max. 99</td></tr> </table>	Name	Länge	Beschreibung	ID	3	Eindeutige Kennung des Artikels	Kennung	1	B ... Basisartikel S ... Subartikel	Anzahl	2	Anzahl des Artikels max. 99
Name	Länge	Beschreibung												
ID	3	Eindeutige Kennung des Artikels												
Kennung	1	B ... Basisartikel S ... Subartikel												
Anzahl	2	Anzahl des Artikels max. 99												
Gültig Ab	16	Start-Zeitpunkt der Gültigkeit im Format ‚DD.MM.YYYY HH:MM‘												
Gültig Bis	16	End-Zeitpunkt der Gültigkeit im Format ‚DD.MM.YYYY HH:MM‘												
Streckencode	3	Bezeichner der Strecke												
Zone Anfang	5	Bezeichner der Streckenanfangszone												
Zone Ende	5	Bezeichner der Streckenendzone												
Zone Via 1	5	Bezeichnung der ersten Via Zone												
Zone Via 2	5	Bezeichnung der zweiten Via Zone. Weitere Via Zonen sind nicht möglich.												

Tabelle 27: Daten Jahreskarte Provider spezifischer Bereich

Die Parameter die mit ‚Efs‘ beginnen sind eine Anleihe aus der Definition des UIC-918-3*-Barcodes durch die DB (Deutsche Bahn). /UIC918-3*/

4 Testbeschreibung

In diesem Kapitel wird der für die Durchführung der Untersuchung notwendige Testaufbau beschrieben sowie die einzelnen Strategien und die durchzuführenden Testfälle.

4.1 Testumgebung

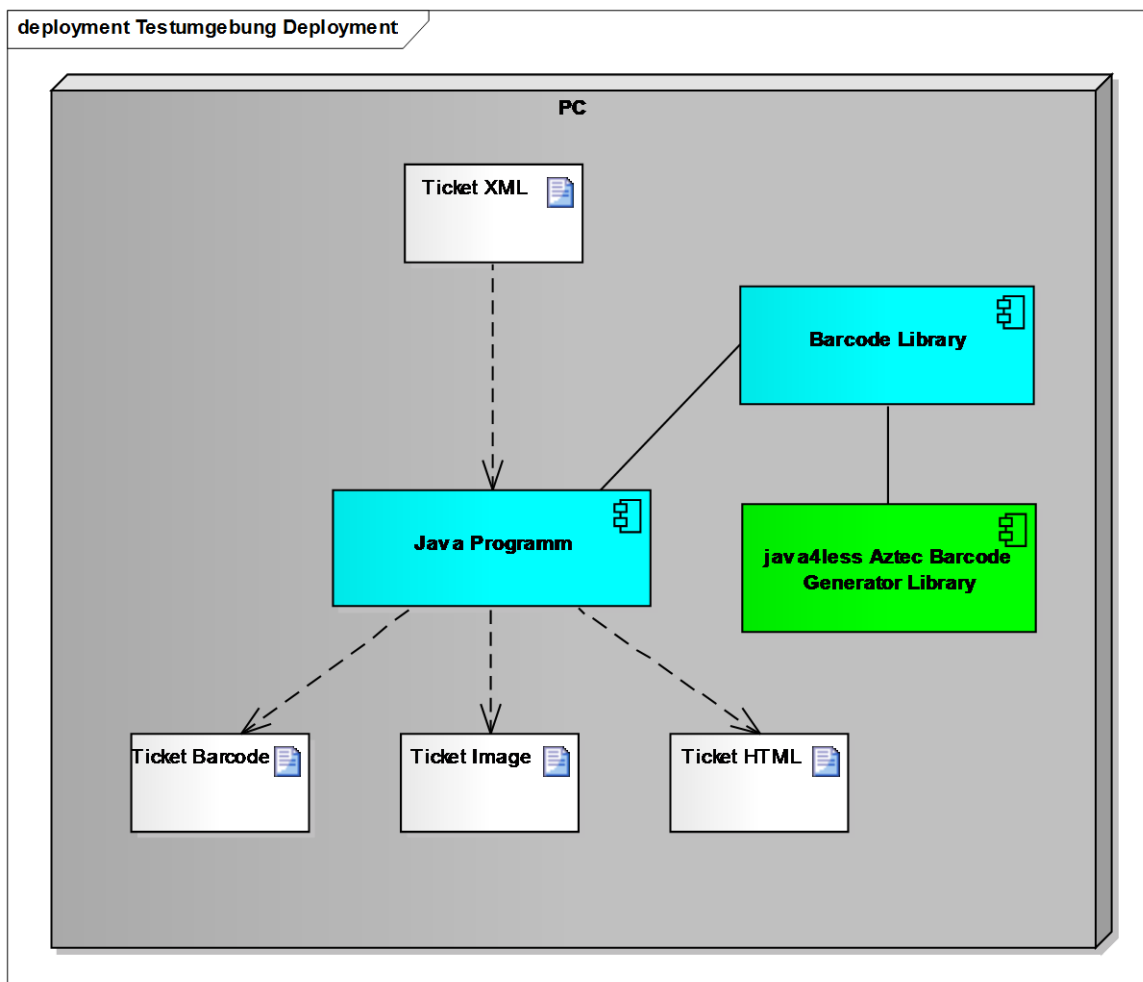


Abbildung 19: Deployment Diagramm Testumgebung

Die folgenden Komponenten werden bei der Testdurchführung verwendet:

Komponente	Beschreibung
PC	Die Tests werden mit Hilfe eines herkömmlichen PC's ohne besondere Anforderungen ausgeführt.
Ticket XML	Ausgangsdatei mit den kompletten Ticketdaten im XML-Format.
Java Programm	Einfaches Java Programm zur Umwandlung des Tickets in einen Barcode, ein Image und für die Darstellung eine HTML-Datei.
Barcode Library	Implementation der Funktion zur Erstellung des Barcodes.
Aztec Barcode Generator Library	Externe Library zur Generierung von Aztec-Barcodes der Firma java4less.
Ticket Barcode	Der UIC-913-3-konforme Barcode im PNG-Format (Portable Network Graphics).
Ticket Image	Die Printversion des ausgestellten Tickets als Image im PNG-Format.
Ticket HTML	Eine HTML-Seite (Hyper Text Markup Language) zur leichteren Darstellung des Ticket-Barcodes und des Ticket-Images.

Tabelle 28: Komponenten der Testumgebung

Exemplarisch folgt nun jeweils ein Beispiel für die Angabe der Eingabe-Parameter als auch die erzeugten Dateien.

4.1.1 Ticket XML

Die Informationen des Tickets werden in Form einer XML-Datei an das Testprogramm übergeben. Diese Datei hat den folgenden Aufbau:

```
<?xml version="1.0" encoding="UTF-8"?>
<TicketUIC9183>
  <!-- Version has to be '01' -->
  <Version>01</Version>
  <!-- RICS 1181 for OEBC -->
  <RICS>3306</RICS>
  <!-- Keyversion has to be 5 digits and the private key must be available -->
  <KeyVersion>TT001</KeyVersion>

  <!-- HEAD Only once in file -->
  <HEAD>
    <!-- unambiguous ticket key (unique id) -->
    <TicketID>1234568909876543210</TicketID>
    <!-- Must be the Format DDMMYYYYHHMM -->
    <TicketTime>311219992359</TicketTime>
    <!-- Flags 1..international Ticket / 2..Edited by Agent / 4..Specimen -->
    <Flags>1</Flags>
    <!-- Language&SecondLanguage 'de' from ISO 639-1
    http://en.wikipedia.org/wiki/List_of_ISO_639-1_codes -->
    <Language>de</Language>
    <SecondLanguage>de</SecondLanguage>
```



```

    <!-- RICS is neccessary at this position and must be the same as the
RICS in the Root node -->
    <RICS>3306</RICS>
</HEAD>

<!-- LayoutRecord only once in File -->
<TLAY>
    <!-- Format of the Record normaly 'RCT2' -->
    <Layout>RCT2</Layout>
    <!-- Version (of the RecordFormat of this Record normaly '01' -->
    <Version>01</Version>
    <!-- Familiennname -->
    <Field>
        <Line>0</Line>
        <Column>53</Column>
        <Height>1</Height>
        <Width>9</Width>
        <Formating>0</Formating>
        <Text>Muster</Text>
    </Field>
    <!-- Vorname -->
    <Field>
        <Line>0</Line>
        <Column>63</Column>
        <Height>1</Height>
        <Width>9</Width>
        <Formating>0</Formating>
        <Text>Max</Text>
    </Field>
    <!-- Gueltig von -->
    <Field>
        <Line>3</Line>
        <Column>1</Column>
        <Height>1</Height>
        <Width>10</Width>
        <Formating>0</Formating>
        <Text>Gültig von</Text>
    </Field>
    <!-- Datum von bis -->
    <Field>
        <Line>3</Line>
        <Column>11</Column>
        <Height>1</Height>
        <Width>37</Width>
        <Formating>0</Formating>
        <Text>28.07.2014 00:01 bis 29.07.2014 23:59</Text>
    </Field>
    <!-- Anzahl der Personen -->
    <Field>
        <Line>1</Line>
        <Column>53</Column>
        <Height>1</Height>
        <Width>11</Width>
        <Formating>0</Formating>
        <Text>Personen 01</Text>
    </Field>
    <!-- Von -->
    <Field>
        <Line>6</Line>

```

```

        <Column>14</Column>
        <Height>1</Height>
        <Width>16</Width>
    <Formating>0</Formating>
        <Text>Wien Westbahnhof</Text>
    </Field>
    <!-- Nach -->
    <Field>
        <Line>6</Line>
        <Column>35</Column>
        <Height>1</Height>
        <Width>16</Width>
        <Formating>0</Formating>
        <Text>St. Pölten</Text>
    </Field>
    <!-- Preis -->
    <Field>
        <Line>13</Line>
        <Column>60</Column>
        <Height>1</Height>
        <Width>13</Width>
        <Formating>0</Formating>
        <Text>EUR 069,70.-</Text>
    </Field>
    <!-- Zusätzlicher Text -->
    <Field>
        <Line>8</Line>
        <Column>2</Column>
        <Height>1</Height>
        <Width>40</Width>
        <Formating>0</Formating>
        <Text>Über Wien Penzing, Wien Hütteldorf</Text>
    </Field>
    <Field>
        <Line>9</Line>
        <Column>2</Column>
        <Height>1</Height>
        <Width>26</Width>
        <Formating>0</Formating>
        <Text>und Purkersdorf Sanatorium</Text>
    </Field>
    <Field>
        <Line>10</Line>
        <Column>2</Column>
        <Height>1</Height>
        <Width>59</Width>
        <Formating>0</Formating>
        <Text>Gilt nur in Verbindung mit einem gültigen Lichtbildaus-
weis</Text>
    </Field>
</TLAY>
</TicketUIC9183>

```

Aus dieser XML-Datei werden die drei folgenden Dateien erzeugt.

4.1.2 Ticket Barcode

Der UIC-918-3-konforme Barcode des Tickets als Image im PNG-Format.



Abbildung 20: UIC-918-3-konformer Aztec-Barcode als Testergebnis

4.1.3 Ticket Image

Die grafische Darstellung des Tickets als Image im PNG-Format.

Fahrschein				Mustern		Max	
Gültig von 15.12.2013 00:00 bis 15.01.2015 23:59				Personen 01			
Datum	Zeit	Von	-> Nach	Datum	Zeit	Klass.	
		Wien Westbahnhof	St. Pölten				
Über Wien Penzing, Wien Hütteldorf und Purkersdorf Sanatorium Gilt nur in Verbindung mit einem gültigen Lichtbildausweis							
				EUR 069,70.-			

Abbildung 21: Darstellung Fahrberechtigung als PNG-Image

4.1.4 Ticket HTML

Zur Übersicht wird zusätzlich eine HTML-Seite erzeugt welche die Darstellung der zusammen gehörigen Barcode- und Ticket-Bilder übernimmt.

```
<html>

<body>



<p/>



</body>
```

4.2 Aufbau Barcode Library

Die für die Erstellung des UIC-918-3-konformen Barcodes notwendigen Funktionen wurden in einer Library gekapselt. Diese Library wurde in der Programmiersprache Java erstellt.

„Doch als robuste objektorientierte Programmiersprache mit einem großen Satz von Bibliotheken ist Java als Sprache für Softwareentwicklung im Großen angekommen und im Bereich plattformunabhängiger Programmiersprachen konkurrenzlos.“ /Javal2011, Abschnitt 1/

Die Library hat folgenden Aufbau:

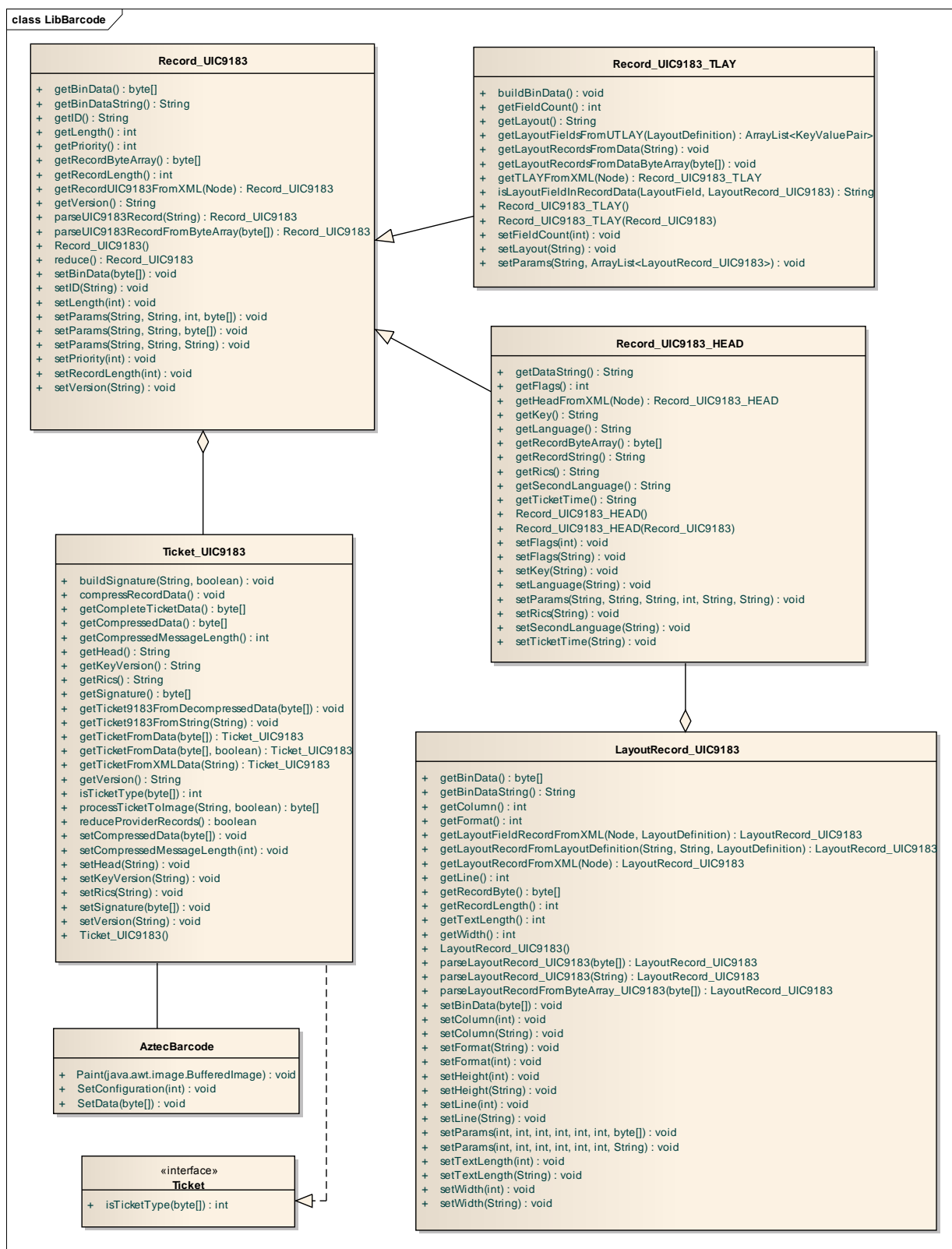


Abbildung 22: Class-Diagramm Barcode-Library

4.3 Kennzahlen

Folgende Daten werden bei der Durchführung der einzelnen Testdurchgänge (Erstellung eines Barcodes nach UIC-918-3) durch die Testsoftware ermittelt:

Name	Einheit	Bemerkung
SizeHeader	Byte	Größe der Datensätze U_HEAD (siehe 2.3.4.4 U_HEAD) und U_TLAY (siehe 2.3.4.5 U_TLAY)
SizeProvData	Byte	Größe der zusätzlich zu transportierenden Daten (siehe 2.3.4.7 Herstellerspezifische Datensätze).
SizeOverall	Byte	Größe der zu komprimierenden Daten (siehe 2.3.4.3 Aufbau Nutzdaten)
SizeCompressed	Byte	Größe der kompletten Nutzdaten nach dem Komprimierungsvorgang (siehe 2.3.4.2 Aufbau gesamter Dateninhalt)
Kompression	Prozent	„We can represent the compression ratio by expressing the reduction in the amount of data required as a percentage of the size of the original data.“ /InDaCo2005, S. 5/ Ermittelt wird der Wert nach der Formel: $\text{Kompression} = 100 - ((\text{Endwert} * 100) / \text{Ausgangswert})$
SizeComplete	Byte	Größe der komplettierten Daten die in dem Aztec Barcode dargestellt werden sollen.
CreateBarcode	Ja/Nein	Konnte ein Barcode nach UIC-918-3 erstellt werden. Sollte kein Barcode erstellt werden können, wird die Anzahl der nicht im Barcode untergebrachten Bits festgehalten.

Tabelle 29: Kennzahlen für die Auswertung

4.4 Strategien

Es wurden die folgenden sechs Strategien zur Abbildung der Daten erarbeitet.

Als gängige Formate zur plattformunabhängigen Speicherung von Daten sind die drei folgenden Formate weit verbreitet:

- XML-Format
- JSON-Format
- YAML-Format

Im Gegensatz zu den vorhergehenden Formaten, ist es möglich selbst Formate zu definieren, welche dann bei Verwendung von der überprüfenden Stelle als Software zu implementieren sind. Die folgenden drei Formate wurden durch den Diplomanden selbst erarbeitet:

- Abbildung als Text
- Abbildung als Text komprimiert
- Proprietäres Format

4.4.1 Abbildung im XML-Format

Die Testdaten stellen sich im XML-Format (beschrieben in 2.8) wie folgt dar:

```
<IN-  
DA><WKZ>E</WKZ><KID>12345678901234567</KID><PID>ABCDEFG</PID><GS>M<  
/GS><GEB>10041969</GEB><NAME>Max Mus-  
ter</NAME><IDT>FS</IDT><IDI>199404123AB</IDI><VFR>28.07.2014  
00:01</VFR><VTO>29.07.2014  
23:59</VTO><STR>STR</STR><ZONE><ZAN>12000</ZAN><ZEN>13488</ZEN><ZV1  
>12745</ZV1><ZV2>13219</ZV2></ZONE></INDA>
```

Der Datenbereich des Barcodes ist vor der Kompression wie folgt, wobei die hinzugefügten Daten gelb gekennzeichnet wurden:

```
U_HEAD01005333061234568909876543210  
3112199923591dedeU_TLAY010391RCT200110053010900006Muster0063010900003M  
ax0301011100011Gültig von031101370003728.07.2014 00:01 bis 29.07.2014  
23:590153011100011Personen 010614011600016Wien Westbahnhof0635011600011St.  
Pölten1360011300012EUR 069,70.-0802014000036Über Wien Penzing, Wien Hüttel-  
dorf0902012600026und Purkersdorf Sanatorium1002015900059Gilt nur in Verbindung  
mit einem gültigen Lichtbildaus-  
weis3306TT010315<INDA><WKZ>E</WKZ><KID>12345678901234567</KID><PID>AB  
CDEFG</PID><GS>M</GS><GEB>10041969</GEB><NAME>Max Mus-  
ter</NAME><IDT>FS</IDT><IDI>199404123AB</IDI><VFR>28.07.2014  
00:01</VFR><VTO>29.07.2014  
23:59</VTO><STR>STR</STR><ZONE><ZAN>12000</ZAN><ZEN>13488</ZEN><ZV1  
>12745</ZV1><ZV2>13219</ZV2></ZONE></INDA>
```


4.4.2 Abbildung im JSON-Format

Eine weitere Möglichkeit die Parameter abzubilden ist das in 2.9 beschriebene Datenformat JSON.

Die Testdaten im JSON-Format stellen sich wie folgt dar:

```
{"GEB":"10041969","STR":"STR","NAME":"Max Muster",  
"IDI":"199404123AB","PID":"ABCDEFGF","ZAN":"12000","ZEN":"13488","VTO":"29.07.2014  
23:59","WKZ":"E","GS":"M","ART":["100B01","113S03"],"IDT":"FS","KID":"12345678901234567",  
"VFR":"28.07.2014 00:01","ZV2":"13219","ZV1":"12745"}
```

Der Datenbereich des Barcodes ist vor der Kompression wie folgt, wobei die hinzugefügten Daten gelb gekennzeichnet wurden:

```
U_HEAD01005333061234568909876543210  
3112199923591dedeU_TLAY010391RCT200110053010900006Muster0063010900003Max  
ax0301011100011Gültig von031101370003728.07.2014 00:01 bis 29.07.2014  
23:590153011100011Personen 010614011600016Wien Westbahnhof0635011600011St.  
Pölten1360011300012EUR 069,70.-0802014000036Über Wien Penzing, Wien Hüttel-  
dorf0902012600026und Purkersdorf Sanatorium1002015900059Gilt nur in Verbindung  
mit einem gültigen Lichtbildaus-  
weis3306TT010286{"GEB":"10041969","STR":"STR","NAME":"Max Mus-  
ter","IDI":"199404123AB","PID":"ABCDEFGF","ZAN":"12000","ZEN":"13488","VTO":"29.07.  
2014  
23:59","WKZ":"E","GS":"M","ART":["100B01","113S03"],"IDT":"FS","KID":"1234567890123  
4567","VFR":"28.07.2014 00:01","ZV2":"13219","ZV1":"12745"}
```

4.4.3 Abbildung im YAML-Format

Eine weitere Möglichkeit die Parameter abzubilden ist das in 2.10 beschriebene Datenformat YAML.

Die Testdaten im YAML-Format stellen sich wie folgt dar:

```
{GEB: '10041969', STR: STR, NAME: Max Muster, IDI: 199404123AB, PID: ABCDEFG,
ZAN: '12000', ZEN: '13488', VTO: '29.07.2014 23:59', GS: M, ART: [999B01, 113S03],
IDT: FS, WZ: E, KID: '12345678901234567', VFR: '28.07.2014 00:01', ZV2: '13219', ZV1:
'12745'}
```

Der Datenbereich des Barcodes ist vor der Kompression wie folgt, wobei die hinzugefügten Daten gelb gekennzeichnet wurden:

```
U_HEAD01005333061234568909876543210
3112199923591dedeU_TLAY010391RCT200110053010900006Muster0063010900003M
ax0301011100011Gültig von031101370003728.07.2014 00:01 bis 29.07.2014
23:590153011100011Personen 010614011600016Wien Westbahnhof0635011600011St.
Pölten1360011300012EUR 069,70.-0802014000036Über Wien Penzing, Wien Hüttel-
dorf0902012600026und Purkersdorf Sanatorium1002015900059Gilt nur in Verbindung
mit einem gültigen Lichtbildausweis3306TT010267{GEB: '10041969', STR: STR, NAME:
Max Muster, IDI: 199404123AB, PID: ABCDEFG, ZAN: '12000', ZEN: '13488', VTO:
'29.07.2014 23:59', GS: M, ART: [999B01, 113S03], IDT: FS, WZ: E, KID:
'12345678901234567', VFR: '28.07.2014 00:01', ZV2: '13219', ZV1: '12745'}
```

4.4.4 Abbildung als Text (SimpleText)

Die einfachste Form der Abbildung der zusätzlichen Daten ist es, die einzelnen Parameter in Textform aneinander zu reihen und mit einem Trennzeichen (zur besseren Analyse der Daten) getrennt abzubilden.

Die folgenden Daten gelten auch als Vorgabe zur exemplarischen Darstellung aller weiteren Strategiebeispiele.

Parametername	Inhalt
Währungskennzeichen	E
KundenId	12345678901234567
PartnerId	ABCDEFGG
EfsFGGeschlecht	M
EfsFGGeb	10041969
EfsFGName	Max Muster
EfsIDTyp	FS
EfsID	199404123AB
Artikel	999B01113S03 (Bestehend aus 2 Artikeln [1*Basisartikel Nummer 999, 3*Subartikel Nummer 113])
Gültig Ab	28.07.2014 00:01
Gültig Bis	29.07.2014 23:59
Streckencode	STR
Zone Anfang	12000
Zone Ende	13488
Zone Via 1	12745
Zone Via 2	13219

Tabelle 30: Beispieldaten zur Darstellung der Strategien

Als Trennzeichen wird das ASCII-Zeichen ‚\‘ (Wert 0x5C) verwendet. Dem entsprechend wird ein zusätzlicher Datensatz mit folgendem Inhalt dem Datenbereich hinzugefügt:

```
E\12345678901234567\ ABCDEFG\ M\10041969\ Max Muster\  
FS199404123AB\100B01113S03\28.07.2014 00:01\29.07.2014 23:59\  
STR\12000\13488\12745\13219
```

Der Datenbereich des Barcodes ist vor der Kompression wie folgt, wobei die hinzugefügten Daten gelb gekennzeichnet wurden:

```
U_HEAD01005333061234568909876543210  
3112199923591dedeU_TLAY010391RCT200110053010900006Muster0063010900003M  
ax0301011100011Gültig von031101370003728.07.2014 00:01 bis 29.07.2014  
23:590153011100011Personen 010614011600016Wien Westbahnhof0635011600011St.  
Pölten1360011300012EUR 069,70.-0802014000036Über Wien Penzing, Wien  
Hütteldorf0902012600026und Purkersdorf Sanatorium1002015900059Gilt nur in  
Verbindung mit einem gültigen  
Lichtbildausweis3306TT010150E\12345678901234567\ABCDEFG\M\10041969\Max  
Muster\Fs199404123AB\100B01113S03\28.07.2014 00:01\29.07.2014  
23:59\STR\12000\13488\12745\13219
```

4.4.5 Abbildung als Text komprimiert (PreCompressed)

Um schon im Vorhinein die Daten möglichst gering zu halten, könnte man die zusätzlich abzuspeichernden Daten schon im Vorfeld komprimieren. Dazu werden die Daten wie in der vorhergehenden Strategie (siehe Abschnitt 4.4.4) dargestellt und im Anschluss mit Hilfe des DEFLATE-Algorithmus komprimiert. Die resultierenden Daten werden als Base64-Darstellung dem Datenbereich hinzugefügt.

Ausgehend von folgendem Datentext:

```
E\12345678901234567\ ABCDEFG\ M\10041969\ Max Muster\  
FS199404123AB\100B01113S03\28.07.2014 00:01\29.07.2014 23:59\  
STR\12000\13488\12745\13219
```

Dieser wird mit Hilfe von DEFLATE komprimiert. Folgend sind die Daten im hexadezimalen Format dargestellt:

```
000>78 9C 3D CD 3B 0E C2 30
008>10 84 E1 AB E4 04 D1 CC
016>EE 3A F6 A6 4B 20 A1 72
024>83 29 B7 A1 E0 02 3C 24
032>8E 8F 91 10 DD A7 5F 23
040>CD 16 14 B5 34 E5 E2 F8
048>29 96 F5 70 DC F6 53 D4
056>20 60 F4 C9 A3 5E DF 43
064>7D 3D 9E B7 7B EC 8D EE
072>D6 BB E8 B2 7E 17 2B 48
080>6A 83 86 94 11 79 14 D0
088>06 60 06 43 FC 1F 44 E7
096>E4 D1 2E E7 7E 08 20 A8
104>56 4A 77 B6 D4 2D F4 0F
112>2F 36 20 E8
```

Die anschließende Umwandlung der Daten in das Base64-Format ergibt:

```
eJw9zTsOwjAQhOGr5ATRzO469qZLIKfygym3oeACPCSOj5EQ3adfl80WFLU05eL4KZb
1cNz2U9QgYPTJo17fQ309nrd77I3u1rvosn4XK0hgg4aUEXkU0AZgBkP8H0Tn5NEu534II
KhWSne21C30Dy82IOg=
```

Der Datenbereich des Barcodes ist vor der finalen Kompression wie folgt, wobei die hinzugefügten Daten gelb gekennzeichnet wurden:

```
U_HEAD01005333061234568909876543210
3112199923591dedeU_TLAY010391RCT200110053010900006Muster0063010900003M
ax0301011100011Gültig von031101370003728.07.2014 00:01 bis 29.07.2014
23:590153011100011Personen 010614011600016Wien Westbahnhof0635011600011St.
Pölten1360011300012EUR 069,70.-0802014000036Über Wien Penzing, Wien Hüttel-
dorf0902012600026und Purkersdorf Sanatorium1002015900059Gilt nur in Verbindung
mit einem gültigen Lichtbildaus-
weis3306TT010168eJw9zTsOwjAQhOGr5ATRzO469qZLIKfygym3oeACPCSOj5EQ3adfl
80WFLU05eL4KZb1cNz2U9QgYPTJo17fQ309nrd77I3u1rvosn4XK0hgg4aUEXkU0AZgBk
P8H0Tn5NEu534IIKhWSne21C30Dy82IOg=
```

4.4.6 Abbildung proprietäres Format (Complex)

Bei dieser Darstellungsform der zusätzlichen Daten wird versucht die einzelnen Parameter so weit wie möglich reduziert darzustellen.

Das Ergebnis wird eine lange Kette von einzelnen Bits sein. Diese Bit-Kette wird danach in eine Anordnung von Bytes umgewandelt, welche schlussendlich in einen Base64-basierenden Text umgewandelt werden.

4.4.6.1 Darstellung Text

Bei der Darstellung von Texten wurde die Huffman-Codierung angewandt. Dazu wurde als Beispieltext die Rede zum Nationalfeiertag von Heinz Fischer (österreichischer Bundespräsident) vom 26.10.2012 ausgewählt. /RedeFischer/

Bei der Analyse der textuellen Parameter der zu erwartenden Ist-Daten wurde ein höheres Aufkommen der Zahlen erkannt. Dem wird Rechnung getragen, indem Zahlen vor der Erstellung des Huffman-Baumes höher priorisiert werden.

Der Huffman-Baum wurde mit Hilfe fremder Softwaremodule erstellt. Dazu wurde der benötigte Quellcode in Java von /HTRosetta/ extrahiert und in einem eigenen Programm verwendet.

Einige textuelle Parameter haben eine fixe Länge. Die anderen mit variabler Länge werden mit einem ,\` Zeichen terminiert. Das Bitmuster für das Zeichen ,\` ist 101100.

Zur besseren Komprimierung der Textdaten wurde der gesamte Text in Großbuchstaben umgewandelt.

Das resultierende Mapping der einzelnen Buchstaben auf Bit-Folgen:

Zeichen	Häufigkeit	Bitmuster	Zeichen	Häufigkeit	Bitmuster
SPACE	824	000	F	214	01101
!	9	101011000	G	216	10000
"	2	101011001110	H	205	111011
,	112	101010	I	213	01100
-	3	10101100101	J	208	00111
.	41	10101101	K	205	111110
0	210	01001	L	211	01010
1	205	111101	M	215	01110
2	205	111100	N	207	111111
3	201	110010	O	201	101101
4	201	110000	P	207	00100
5	202	110110	Q	201	110001
6	203	111000	R	211	01011
7	203	111001	S	217	10001
8	201	101111	T	203	110111
9	202	110101	U	215	01111
:	4	10101100110	V	208	00101
;	1	101011001000	W	208	00110
?	83	1010111	X	201	110011
A	218	10011	Y	201	101110
B	209	01000	Z	205	111010
C	202	110100	[1	101011001001
D	223	10100	\	200	101100
E	217	10010]	1	1010110011110

Tabelle 31: Huffman-Code-Mapping Test

Dem entsprechend werden die textuellen Parameter im Beispiel wie folgt umgewandelt:

Parametername	Inhalt
Währungskennzeichen	E
10010	
KundenId	12345678901234567
11110111110011001011000011011011100011100110111111010101001111101111100	
110010110000110110111000111001101100	
PartnerId	ABCDEFGH
100110100011010010100100100110110000101100	
EfsFGGeschlecht	M
01110	
EfsFGName	MAX MUSTER
01110100111100110000111001111100011101111001001011101100	
EfsIDTyp	FS
0110110001101100	
EfsID	199404123AB
111101110101110101110000010011100001111011111001100101001101000101100	
Artikel	999B01113S03
1101011101011101010100001001111101111011110111100101000101001110010101	
100	
Streckencode	STR
1000111011101011101100	

Tabelle 32: Parameter Texte als Huffman-Code Bitmuster

4.4.6.2 Darstellung Zahlen

Parameter die nur durch natürliche Zahlen abgebildet werden können, werden in einen 16 Bit Binärcode umgewandelt:

Parametername	Inhalt	BitCode
Zone Anfang	12000	0010111011100000
Zone Ende	13488	0011010010110000
Zone Via 1	12745	0011000111001001
Zone Via 2	13219	0011001110100011

Tabelle 33: Parameter natürliche Zahlen als Bitmuster

4.4.6.3 Darstellung Datum

Zur Darstellung einzelner Datumsangaben wurden jeweils die Differenztage zu einem gewissen Datum gebildet und diese Anzahl in einen 16 Bit Binärcode umgewandelt.

Parametername	Inhalt	Referenzdatum	Differenztage	Bitmuster
EfsFGGeb	10.04.1969	01.01.1900	25301	0110001011010101
Gültig Ab	28.07.2014	01.01.2000	5322	0001010011001010
Gültig Bis	29.07.2014	01.01.2000	5323	0001010011001011

Tabelle 34: Parameter Datum als Bitmuster

4.4.6.4 Darstellung Zeit

Bei den Anforderungen haben nur die Gültigkeitsgrenzen eine Tageszeit als Teil des Parameters. Die Tageszeit wird als Differenzminuten zur Mitternacht errechnet und als 16 Bit Binärcode dargestellt. Benötigt wären eigentlich nur 11 Bit (1439 Minuten maximal möglich), zur leichten Darstellung wurden aber 16 Bit gewählt.

Parametername	Inhalt	Differenzminuten	Bitmuster
Gültig Ab	00:01	1	0000000000000001
Gültig Bis	23:59	1439	0000010110011111

Tabelle 35: Parameter Uhrzeit als Bitmuster

Dem entsprechend werden die Parameter „Gültig Ab“ und „Gültig Bis“ als Zusammensetzung des Datums- und des Zeitwertes dargestellt:

Parametername	Inhalt	Bitmuster
Gültig Ab	28.07.2014 00:01	00010100110010100000000000000001
Gültig Bis	29.07.2014 23:59	00010100110010110000010110011111

Tabelle 36: Parameter „Gültig Ab“ und „Gültig Bis“ als Bitmuster

4.4.6.5 Bildung Bit-Kette und Base64-Text

Um die Darstellung nun zu vervollständigen, werden die Bitmuster der einzelnen Parameter einfach hintereinander geschrieben.

Daraus ergibt sich folgende Bit-Kette:

```
10010111101111100110010110000110110111000111001101111110101010011111011
11100110010110000110110111000111001101100100110100011010010100100100110
11000010110001110011101001111001100001110011111000111011110010010111011
00011000101101010101101100011011001111011101011101011100000100111000011
110111110011001010011010001011000001010011001001000000000000000010001010
01100101100000101100111111000111011101011101100001011101110000000110100
10110000001100011100100100110011101000111101011101011101010100001001111
1011111011111011100101000101001110010101100
```

Bei der Umwandlung der Bit-Kette wird diese in Abschnitte der Länge von 8 Bit geteilt. Es kann durchaus vorkommen, dass eine Anzahl an Bits (kleiner als 8) am Ende über bleiben. Diese werden auch in ein Byte umgewandelt und am Ende wird eine noch die Anzahl der verwendeten Bits angehängt. In diesem Fall sind das 4 Bits und die Zahl 4 wurde am Ende dazu gefügt.

Diese Bit-Kette wird danach in eine Anordnung an Bytes umgewandelt (Darstellung der Daten im hexadezimalen Format):

```
0>97 BE 65 86 DC 73 7E A9
8>F7 CC B0 DB 8E 6C 9A 34
16>A4 9B 0B 1C E9 E6 1C F8
24>EF 25 D8 C5 AA D8 D9 EE
32>BA E0 9C 3D F3 29 A2 C1
40>4C 90 00 11 4C B0 59 F8
48>EE BB 0B B8 0D 2C 0C 72
56>4C E8 F5 D7 54 27 DF 7D
64>CA 29 CA 0C 04
```

Final wird die Byte-Anordnung in einen Base64-codierten Text umgewandelt:

```
I75lhtxzfqn3zLDbjmyaNKSbCxzp5hz47yXYxarY2e664Jw98ymiwUyQABFMsFn47rsLuA0s
DHJM6PXXVCffcopygwE
```

Der Datenbereich des Barcodes ist vor der Kompression wie folgt, wobei die hinzugefügten Daten gelb gekennzeichnet wurden:

U_HEAD01005333061234568909876543210
3112199923591dedeU_TLAY010391RCT200110053010900006Muster0063010900003M
ax0301011100011Gültig von031101370003728.07.2014 00:01 bis 29.07.2014
23:590153011100011Personen 010614011600016Wien Westbahnhof0635011600011St.
Pölten1360011300012EUR 069,70.-0802014000036Über Wien Penzing, Wien Hüttel-
dorf0902012600026und Purkersdorf Sanatorium1002015900059Gilt nur in Verbindung
mit einem gültigen Lichtbildaus-
weis3306TT010104I75lhtxzfqn3zLDbjmyaNKSbCxzp5hz47yXYxarY2e664Jw98ymiwUyQ
ABFMsFn47rsLuA0sDHJM6PXXVCffcopygwE

4.5 Testdaten

Die Testdaten wurden auf Basis moderner Softwaretest-Verfahren ermittelt. Zuerst wurden die Daten in Äquivalenzklassen abgebildet.

„Zu einer Äquivalenzklasse gehören alle Eingabedaten, bei denen der Tester davon ausgeht, dass sich das Testobjekt bei Eingabe eines beliebigen Wertes aus der Äquivalenzklasse gleich verhält.“ /SWT2005, S. 109/

Zusätzlich wurde dann auf die bereits ermittelten Daten die Grenzwertanalyse angewandt.

„Bei der Grenzwertanalyse werden die >>Ränder<< der Äquivalenzklassen einer Überprüfung unterzogen. ... Ein Test mit Grenzfällen deckt daher oft Fehlerwirkungen auf.“ /SWT2005, S. 121/

Dem entsprechend ergeben sich die folgenden fünf Testfälle:

- Minimale Daten für Jahreskarte
- Maximale Daten für Jahreskarte
- Maximale Daten nach Spezifikation
- Bestehende Jahreskarte
- Unterschiedliche Daten im erweiterten Datenbereich

Zusätzlich, zu den oben angegebenen Testfällen wird jeweils pro Testfall ein Durchgang ohne erweiterbare Daten durchgeführt.

4.5.1 Minimale Daten für Jahreskarte (Minimal)

Bei diesem Testfall werden die Werte der Datenfelder mit so geringen Daten wie möglich angenommen:

Name	Max Länge	Wert		
Vorname	10	A		
Familienname	10	B		
Von	20	Platt		
Nach	20	Velm		
Gültig Ab	16	28.07.2014 00:01		
Gültig Bis	16	29.07.2014 00:01		
Preis	8	001.00.-		
Zusätzlicher Text	200	entfällt		
Anzahl der Personen	1	1		
Währungskennzeichen	1	E		
KundenId	18	1		
PartnerId	10	1		
EfsFGGeschlecht	1	M		
EfsFGGeb	8	01011990		
EfsFGName	25	A B		
EfsIDTyp	2	FS		
EfsID	10	1		
Artikel	6/Artikel	ID	Kennung	Anzahl
		001	B	01
Gültig Ab	16	28.07.2014 00:01		
Gültig Bis	16	29.07.2014 00:01		
Streckencode	3	001		
Zone Anfang	5	00725		
Zone Ende	5	00511		
Zone Via 1	5	entfällt		
Zone Via 2	5	entfällt		

Tabelle 37: Testdaten des Testfalls minimale Daten für Jahreskarte

4.5.2 Maximale Daten für Jahreskarte (Maximal)

Bei diesem Testfall werden die Werte der Datenfelder mit so vielen Daten wie möglich, entsprechend der Sinnhaftigkeit der Jahreskarte, angenommen:

Name	Max Länge	Wert		
Vorname	10	Antoinella		
Familienname	10	Mustermann		
Von	20	Hetzmannsdorf-Wuller		
Nach	20	Enzersdf. bei Staatz		
Gültig Ab	16	28.07.2014 13:42		
Gültig Bis	16	29.12.2014 20:59		
Preis	8	123.89.-		
Zusätzlicher Text	200	Über Leobendorf Über Langenzersdorf Über Floridsdorf Im Preis enthalten sind 20% Mehrwertsteuer		
Anzahl der Personen	1	9		
Währungskennzeichen	1	E		
KundenId	18	123456789012345678		
PartnerId	10	0987654321		
EfsFGGeschlecht	1	W		
EfsFGGeb	8	31021987		
EfsFGName	25	Antoinella Mustermann		
EfsIDTyp	2	FS		
EfsID	10	ABCDEFHIJK		
Artikel	6/Artikel	ID	Kennung	Anzahl
		001	B	01
		102	S	24
		346	S	35
		978	S	68
Gültig Ab	16	28.07.2014 13:42		
Gültig Bis	16	29.12.2014 20:59		
Streckencode	3	001		
Zone Anfang	5	00635		
Zone Ende	5	00715		
Zone Via 1	5	00425		
Zone Via 2	5	00220		

Tabelle 38: Testdaten des Testfalls maximale Daten für Jahreskarte

4.5.3 Maximale Daten nach Spezifikation (MaximalSpec)

Bei diesem Testfall werden die Werte der Datenfelder mit so vielen Daten wie möglich, entsprechend der Spezifikation der Datenfelder angenommen, auch wenn dies bei der zukünftigen Erstellung der Jahreskarten nicht vorkommen wird:

Name	Max Länge	Wert		
Vorname	10	Antoinella		
Familienname	10	Mustermann		
Von	20	Hetzmannsdorf-Wuller		
Nach	20	Enzersdf. bei Staatz		
Gültig Ab	16	28.07.2014 13:42		
Gültig Bis	16	29.12.2014 20:59		
Preis	8	123.89.-		
Zusätzlicher Text	200	„In unserem Land, das heute seinen Nationalfeiertag begeht, leben mehr als acht Millionen Menschen in neun Bundesländern und in mehr als 2.300 Gemeinden. Männer und Frauen, Ältere und Jüngere; aus v“ (Text mit 200 Buchstaben aus der Rede zum Nationalfeiertag von Heinz Fischer (österreichischer Bundespräsident) vom 26.10.2012 /RedeFischer/)		
Anzahl der Personen	1	9		
Währungskennzeichen	1	E		
KundenId	18	123456789012345678		
PartnerId	10	0987654321		
EfsFGGeschlecht	1	W		
EfsFGGeb	8	31021987		
EfsFGName	25	Antoinella Mustermann1435		
EfsIDTyp	2	FS		
EfsID	10	ABCDEFHIJK		
Artikel	6/Artikel	ID	Kennung	Anzahl
		001	B	01
		102	S	24
		346	S	35
		978	S	68
Gültig Ab	16	28.07.2014 13:42		
Gültig Bis	16	29.12.2014 20:59		
Streckencode	3	986		
Zone Anfang	5	00635		
Zone Ende	5	00715		

Zone Via 1	5	00425
Zone Via 2	5	00220

Tabelle 39: Testdaten des Testfalls maximale Daten nach Spezifikation

4.5.4 Bestehende Jahreskarte (BestehendeJK)

Bei diesem Testfall werden die Werte der Datenfelder durch eine bestehende und aktuell vergebene Jahreskarte bestimmt:

Name	Max Länge	Wert		
Vorname	10	Gregor		
Familiennamen	10	Czanitz		
Von	20	Paasdorf		
Nach	20	Wien		
Gültig Ab	16	01.12.2013 00:01		
Gültig Bis	16	31.01.2015 23:59		
Preis	8	345.10.-		
Zusätzlicher Text	200	Über Auersthal		
Anzahl der Personen	1	1		
Währungskennzeichen	1	E		
KundenId	18	745-344-12		
PartnerId	10	01055		
EfsFGGeschlecht	1	M		
EfsFGGeb	8	17031965		
EfsFGName	25	Gregor Czanitz		
EfsIDTyp	2	FS		
EfsID	10	0727863-BH		
Artikel	6/Artikel	ID	Kennung	Anzahl
		099	B	01
Gültig Ab	16	01.12.2013 00:01		
Gültig Bis	16	31.01.2015 23:59		
Streckencode	3	365		
Zone Anfang	5	00515		
Zone Ende	5	00510		
Zone Via 1	5	00310		
Zone Via 2	5	entfällt		

Tabelle 40: Testdaten des Testfalls bestehende Jahreskarte

4.5.5 Unterschiedliche Daten im erweiterten Datenbereich (Different-Data)

Viele der Lösungsstrategien bauen auf die möglichst effiziente Komprimierung der Daten, welche vorliegt wenn es mehrere gleiche Daten gibt. Zum Beispiel können gleiche Angaben im Feld Vorname und im Feld EfsFGName gut komprimiert werden. Dieser Testfall soll abklären mit welcher Lösungsstrategie Daten am sinnvollsten eingebunden werden können, wenn diese nicht den Daten aus dem bedruckbaren Bereich des Tickets (in diesem Fall die Jahreskarte) entsprechen. Basis ist der Testfall „bestehende Jahreskarte“.

Name	Max Länge	Wert		
Vorname	10	Gregor		
Familienname	10	Czanitz		
Von	20	Paasdorf		
Nach	20	Wien		
Gültig Ab	16	01.12.2013 00:01		
Gültig Bis	16	31.01.2015 23:59		
Preis	8	345.10.-		
Zusätzlicher Text	200	Über Auersthal		
Anzahl der Personen	1	1		
Währungskennzeichen	1	E		
KundenId	18	745-344-12		
PartnerId	10	01055		
EfsFGGeschlecht	1	M		
EfsFGGeb	8	17031965		
EfsFGName	25	Roman Stephan Friedrich		
EfsIDTyp	2	FS		
EfsID	10	0727863-BH		
Artikel	6/Artikel	ID	Kennung	Anzahl
		099	B	01
Gültig Ab	16	14.05.1999 15:46		
Gültig Bis	16	29.11.2014 20:37		
Streckencode	3	365		
Zone Anfang	5	00515		
Zone Ende	5	00510		
Zone Via 1	5	00310		
Zone Via 2	5	Entfällt		

Tabelle 41: Testdaten des Testfalls unterschiedliche Daten im erweiterten Datenbereich

5 Auswertung Ergebnisse

In diesem Kapitel werden die Ergebnisse der Testdurchführung analysiert. Im Abschnitt 5.1 werden alle aufgenommenen Kennzahlen (Beschreibung der Kennzahlen siehe Abschnitt 4.3) tabellarisch dargestellt.

Zur Übersicht sind in der Spalte „CreateBarcode“ jene Felder rot markiert um darzustellen welche Testfälle nicht erfolgreich durchgeführt werden konnten (d. h.: es konnte kein Aztec-Barcode erzeugt werden). Zusätzlich, wie in Tabelle 29: Kennzahlen für die Auswertung beschrieben, wird die Anzahl der verbleibenden Bits bei nicht erfolgreich durchgeführten Testfällen dargestellt.

Es wurde entsprechend der Beschreibung der Testumgebung (siehe Abschnitt 4.1) für jede Strategie (siehe Abschnitt 4.4) in Kombination mit jeweils allen Testdaten (siehe Abschnitt 4.5) eine Darstellung im XML-Format erstellt (siehe Abschnitt 4.1.1). Es wurde zu jeder Strategie eine Fahrberechtigung ohne zusätzliche Daten zum zusätzlichen Vergleich erzeugt „NoProvData“. Das ergibt 35 durchgeführte Testfälle.

Jeder der 35 Testfälle wurde manuell mit Hilfe der Testumgebung (im speziellen unter der Benutzung der BarcodeLibrary siehe Abschnitt 4.2) durchgeführt und die Kennzahlen ermittelt.

In den nach Abschnitt 5.1 folgenden Abschnitten werden die Ergebnisse und somit die einzelnen Strategien einander gegenübergestellt.

5.1 Übersicht Testergebnisse

		Strategie	SizeHeader	SizeProvData	SizeOverall	SizeCompressed	Kompression	SizeComplete	Create Barcode
Testfall	Minimal	NoProvData	231	0	231	163	29,44	231	Ja
		SimpleText	231	97	328	210	35,98	278	Ja
		PreCompressed	231	104	335	257	23,28	325	Ja
		Complex	231	64	295	219	25,76	287	Ja
		XML	231	286	517	313	39,46	381	Ja
		JSON	231	228	459	282	38,56	350	Ja
		YAML	231	216	447	280	37,36	348	Ja
	Maximal	NoProvData	427	0	427	293	31,38	361	Ja
		SimpleText	427	178	605	379	37,36	447	Ja
		PreCompressed	427	196	623	462	25,84	530	862
		Complex	427	128	555	403	27,39	471	Ja
		XML	427	416	843	503	40,33	571	1472
		JSON	427	318	745	461	38,12	529	872
		YAML	427	301	728	460	36,81	528	612

Tabelle 42: Testergebnisse für die Testfälle Minimal und Maximal

		Strategie	SizeHeader	SizeProvData	SizeOverall	SizeCompressed	Kompression	SizeComplete	Create Barcode
Testfall	MaximalSpec	NoProvData	494	0	494	356	27,94	424	Ja
		SimpleText	494	182	676	446	34,02	514	532
		PreCompressed	494	200	694	528	23,92	596	1912
		Complex	494	132	626	467	25,40	535	672
		XML	494	420	914	569	37,75	637	2012
		JSON	494	322	816	526	35,54	594	1972
		YAML	494	305	799	522	34,67	590	1502
	BestehendeJK	NoProvData	272	0	272	205	24,63	273	Ja
		SimpleText	272	135	407	270	33,66	338	Ja
		PreCompressed	272	160	432	338	21,76	406	Ja
		Complex	272	96	368	286	22,28	354	Ja
		XML	272	329	601	382	36,44	450	Ja
		JSON	272	266	538	349	35,13	417	Ja
		YAML	272	250	522	345	33,91	413	Ja
	DifferentData	NoProvData	272	0	272	205	24,63	273	Ja
		SimpleText	272	144	416	296	28,85	364	Ja
		PreCompressed	272	168	440	344	21,82	412	Ja
		Complex	272	108	380	296	22,11	364	Ja
		XML	272	338	610	409	32,95	477	52
		JSON	272	275	547	377	31,08	445	Ja
		YAML	272	259	531	373	29,76	441	Ja

Tabelle 43: Testergebnisse für die Testfälle MaximalSpec, BestehendeJK und DifferentData

5.2 Vergleich „SimpleText“ und „PreCompressed“

Im direkten Vergleich der beiden Strategien „SimpleText“ und „PreCompressed“ bringt die vorherige Komprimierung der hinzuzufügenden Daten keine Vorteile. Im Gegenteil. Durch die Abbildung der hinzuzufügenden Daten als komprimierte Daten im Base64 Format, kann der Kompressionsalgorithmus DEFLATE den gesamten Datenbereich nicht so effizient komprimieren.

Zusätzlich sind die im Vorfeld komprimierten und im Base64-Format dargestellten Daten auch in jedem der Testfälle größer als die Ursprungsdaten. Der Grund hierfür sind die Verwaltungsdaten die innerhalb des komprimierten Datenbereichs abgespeichert werden müssen.

Dem entsprechend ist von der Darstellung zusätzlicher Daten innerhalb eines UIC-918-3-konformen Barcodes nach der Strategie „PreCompressed“ abzuraten.

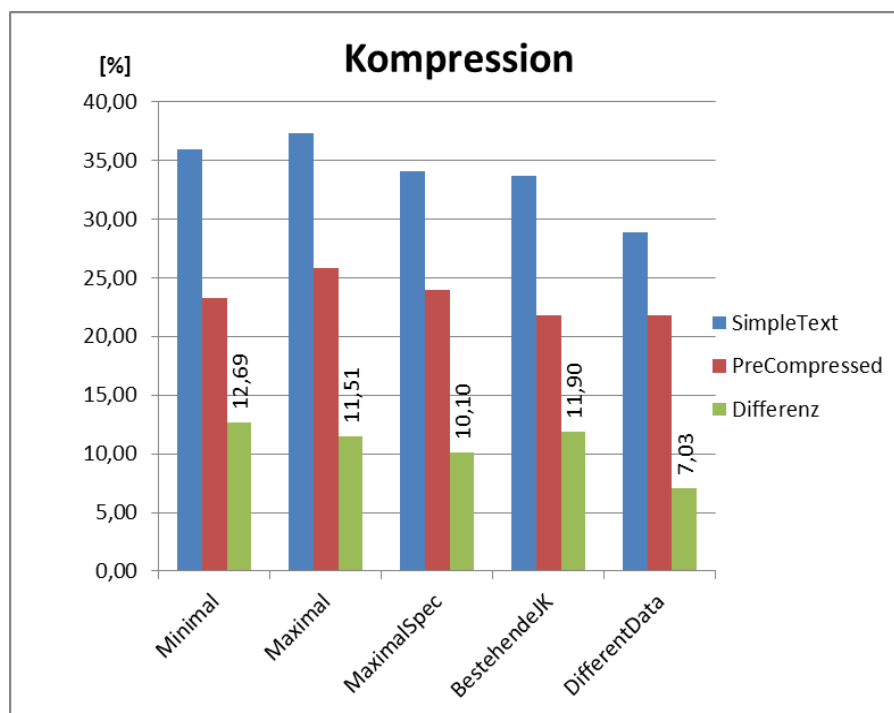


Abbildung 23: Kompression-Strategie „SimpleText“ und „PreCompressed“

5.3 Vergleich Datenaustauschformate XML, JSON und YAML

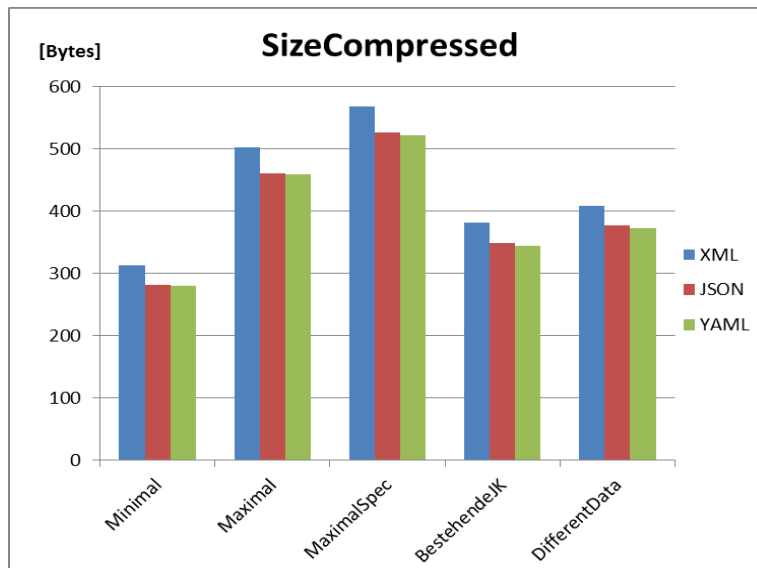


Abbildung 24: Größe komprimierte Daten – Strategie XML, JSON und YAML

Bei den standardisierten Datendarstellungsformaten zeigt sich, dass die Datendarstellung im YAML-Format aufgrund der deutlich reduzierten Steuerzeichen das geringste Datenaufkommen erzeugt.

Auf die Kompression hat das Format nur einen geringfügigen Einfluss, wie das folgende Diagramm zeigt. So kann man dem folgenden Diagramm entnehmen, dass sich Daten im XML-Format stärker komprimieren lassen (höhere Kompressionsrate).

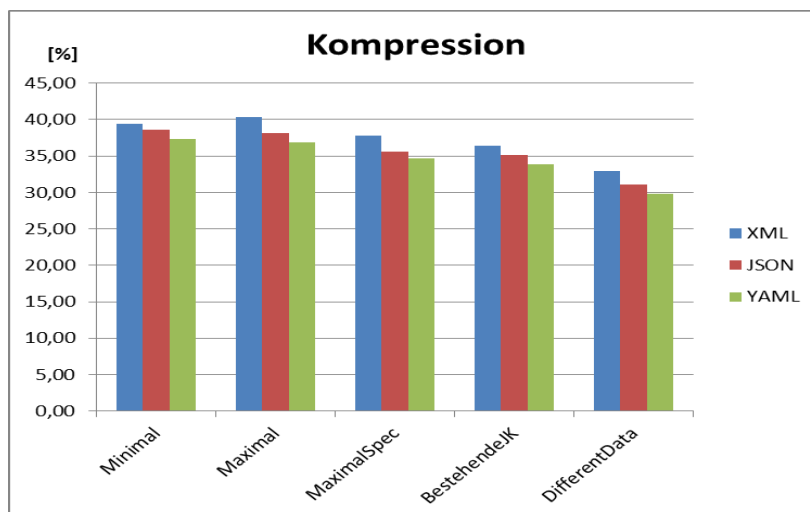


Abbildung 25: Kompression-Strategie XML, JSON und YAML

5.4 Vergleich „SimpleText“ und „Complex“

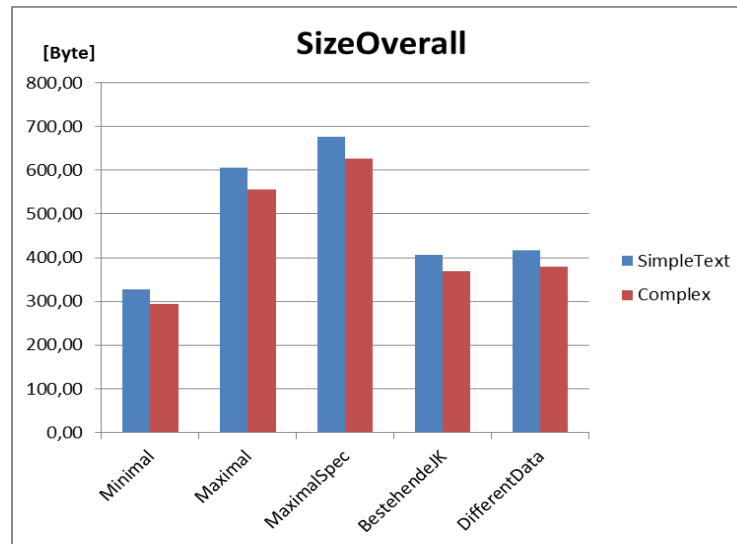


Abbildung 26: Datengröße vor der Kompression-Strategie „SimpleText“ und „Complex“

Bei der Strategie „Complex“ ist die Datengröße vor der Kompression kleiner als bei der Strategie „SimpleText“, wie die Abbildung 26 zeigt. Im folgenden Diagramm erkennt man jedoch, dass sich im Endeffekt ein Vorteil nach der Kompression bei der Strategie „SimpleText“ ergibt.

Als Nachteil für die Implementation nach der Strategie „Complex“ gelten die zusätzlichen Entwicklungsaufwände und daraus resultierend noch zusätzliche Fehlerquellen.

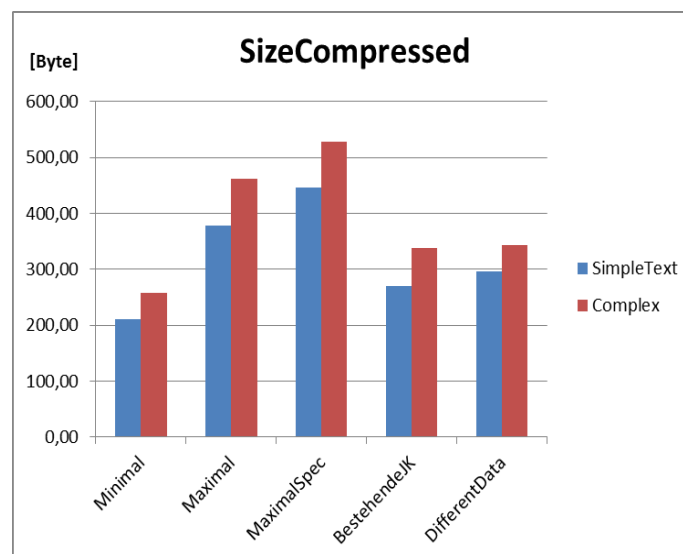


Abbildung 27: Datengröße nach der Kompression-Strategie „SimpleText“ und „Complex“

5.5 Vergleich „SimpleText“ und „YAML“

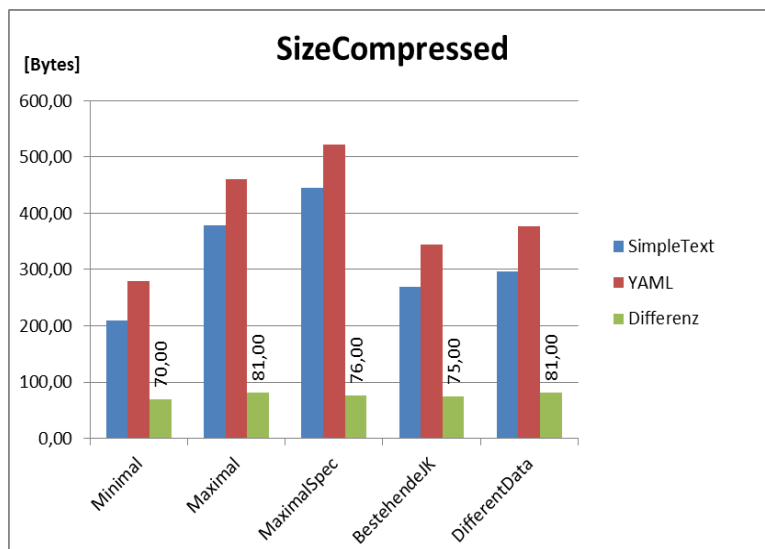


Abbildung 28: Datengröße nach der Kompression-Strategie „SimpleText“ und „YAML“

Nach dem sich die Datendarstellung im YAML-Format als beste standardisierte Darstellung herauskristallisiert hat, wird diese Darstellung nun direkt mit der effizientesten, durch den Diplomanden erstellten, Strategie verglichen.

Der direkte Vergleich zeigt, dass sich bei der Datendarstellung im YAML-Format eine höhere Kompressionsrate erzielen lässt (Abbildung 29: Kompression-Strategie „SimpleText“ und „YAML“). Dieser Vorteil wird jedoch durch die geringere Anzahl an Steuerzeichen (im Falle der Strategie „SimpleText“ Trennzeichen) wieder aufgehoben. Dies resultiert in einer geringeren Datengröße bei der Strategie „SimpleText“.

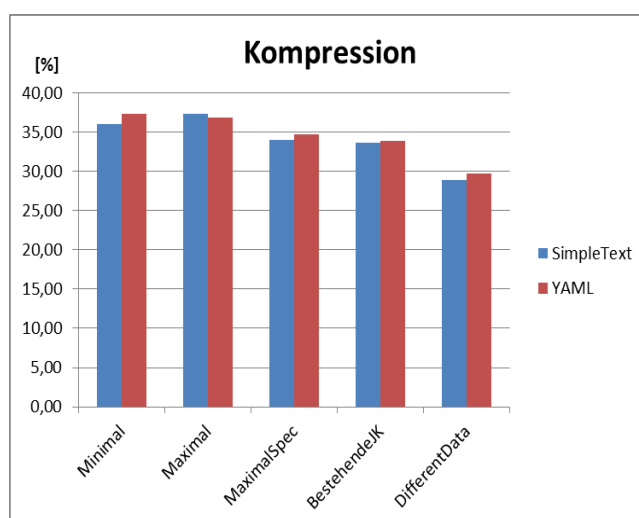


Abbildung 29: Kompression-Strategie „SimpleText“ und „YAML“

5.6 Vergleich „BestehendeJK“ und „DifferentData“

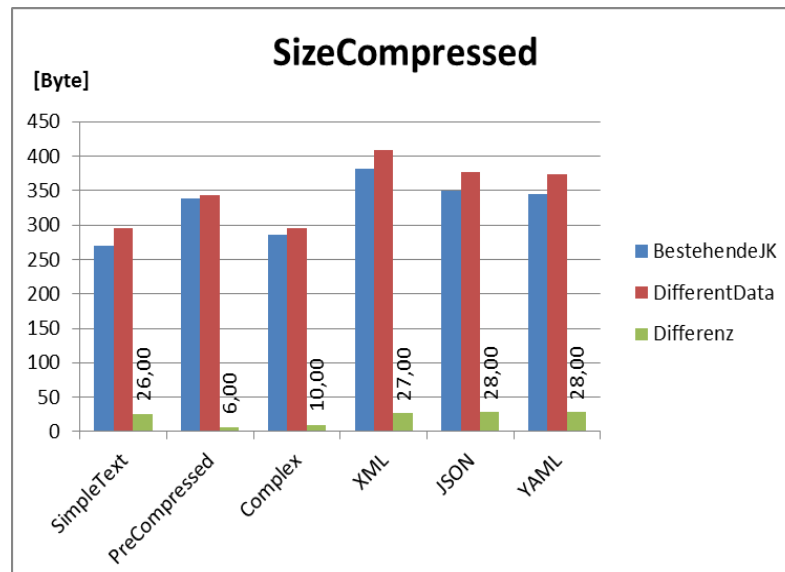


Abbildung 30: SizeCompressed-Strategie „BestehendeJK“ und „DifferentData“

Sollten Daten dem UIC-918-3-konformen Barcode hinzugefügt werden, welche nicht auf das Ticket aufgedruckt werden, sollte man keine der standardisierten Darstellungsformen verwenden. Hier zeigt die Strategie „Complex“ das geringste Datenaufkommen.

Durch die unterschiedlichen Daten ergibt sich eine schlechtere Kompressionsrate, welche sich in einem größeren Datenaufkommen zeigt. Nur bei den beiden selbst definierten Darstellungsformen die nicht auf Datendarstellung im Textformat basieren, zeigen sich nahezu die gleiche Kompressionsraten.

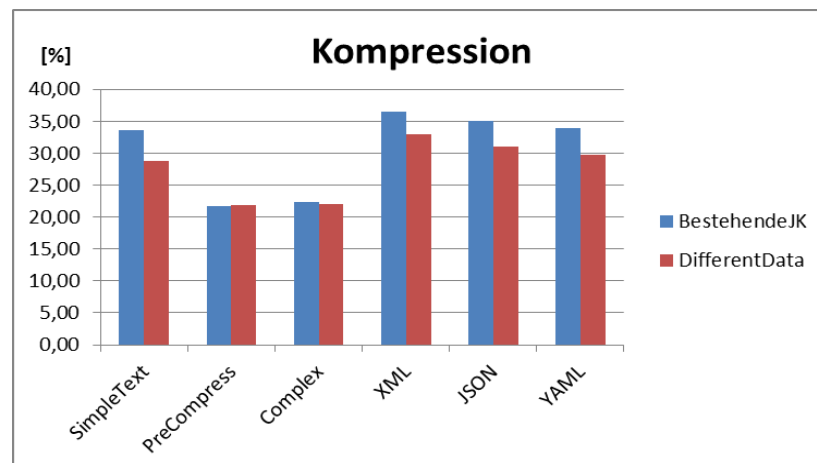


Abbildung 31: Kompression-Strategie „BestehendeJK“ und „DifferentData“

6 Zusammenfassung und Ausblick

In diesem Kapitel werden die Testergebnisse zusammengefasst und die beste Lösung für die Aufgabenstellung durch den Diplomanden erarbeitet.

Da es sich um eine reale Aufgabenstellung handelt wird anschließend in Abschnitt 6.2 die produktiv eingesetzte Lösung beschrieben sowie ein möglicher zukünftiger Ausblick gegeben.

6.1 Zusammenfassung Testergebnisse

Es wurden mit Hilfe des, durch den Diplomanden erstellten, Testaufbaus die einzelnen Strategien zur Übermittlung zusätzlicher Daten innerhalb eines UIC-918-3-konformen Barcodes unter Einhaltung der vorgegebenen Richtlinien (siehe 3.1 Anforderungen durch die Vorschrift UIC-918-3) überprüft.

Die dafür notwendigen Abläufe wurden anhand des durch den Diplomanden entwickelten Prozesses (siehe 3.1 Anforderungen durch die Vorschrift UIC-918-3) in einem Programm (siehe 4.2 Aufbau Barcode Library) realisiert.

Die einzelnen überprüften Strategien sind entweder bestehende und weit verbreitete Darstellungsformen von Daten oder wurden durch den Diplomanden entwickelt (siehe Abschnitt 4.4.4 bis Abschnitt 4.4.6). Im Besonderen wurde bei der Darstellungsform Complex (siehe 4.4.6 Abbildung proprietäres Format (Complex)) eine optimal speicherschonende Darstellungsform der Ticketdaten in Anlehnung an die Huffman-Codierung (siehe 2.6.2 Huffman-Codierung) durch den Diplomanden entwickelt.

Anhand der Testergebnisse lassen sich folgende Aussagen treffen:

- Die beste Strategie für die Realisierung der Jahreskarte des VOR ist die Darstellung als Text mit einem Trennzeichen (siehe 4.4.4 Abbildung als Text (Simple-Text)).
- Die Aufgabenstellung lässt sich nicht vollständig mit allen angegebenen Strategien realisieren. Es konnten einige Barcodes anhand von gewissen Strategien nicht erzeugt werden.
- Sollte in den zusätzlich zu übermittelnden Daten hauptsächlich Parameter enthalten sein, welche nicht im druckbaren Bereich der Daten enthalten sind, dann ist es sinnvoll eine spezielle Strategie dafür zu entwickeln (siehe 5.6 Vergleich „BestehendeJK“ und „DifferentData“).

6.2 Eingesetzte Lösung zur Aufgabenstellung und Ausblick

Auf Basis dieser Diplomarbeit wurde in mehreren Arbeitstreffen mit dem VOR folgende Vorgaben zur Realisierung der Jahreskarte als UIC-918-3-konformer Barcode entwickelt:

- Die zusätzlich zu übermittelnden Daten werden als Standard-Text angegeben.
- Da es für die Erstellung des Barcodes keine fixe und leicht zu überprüfende Grenze gibt, ist die Möglichkeit zu berücksichtigen, dass eine Jahreskarte nicht darstellbar ist.
- Es werden die zusätzlich zu übermittelnden Daten anhand von Themenkreisen (Fahrinformation, Kundenangaben und Streckeninformation) in einzelne Datensätze zusammengefasst. Diesen Datensätzen wird eine Priorität anhand ihrer Wichtigkeit bei der Kontrolle der Fahrtberechtigung zugeordnet. Sollte bei der Erstellung eine Jahreskarte nicht produziert werden können, wird der Datensatz mit der niedrigsten Priorität gelöscht und danach ein neuer Versuch (entsprechend des Prozesses siehe Abschnitt 3.1 Anforderungen durch die Vorschrift UIC-918-3) gestartet.

Es hat sich jedoch anhand der Testdurchgänge gezeigt, dass eine Übermittlung zusätzlicher Daten innerhalb des UIC-918-3-konformen Barcodes, aufgrund des begrenzten Datenvolumens, nicht sonderlich effizient ist.

Eine mögliche alternative Herangehensweise zur Übermittlung von zusätzlichen Daten wird in der Vorschrift UIC-918-3* /UIC918-3*/ durch die DB aufgezeigt. UIC-918-3* baut auf UIC-918-3 auf, lässt aber sämtliche aufgedruckte Informationen innerhalb des erstellten Barcodes weg. Es gibt nur den herstellerspezifischen Datenteil. Dadurch ist die Fahrtberechtigung nicht mehr UIC-918-3-konform und bei einer Validierung kann keine Sichtkontrolle zwischen der ausgedruckten Fahrtberechtigung und der über den Aztec-Barcode abgebildeten Fahrtberechtigung durchgeführt werden. Dem entsprechend ist ein zusätzlicher Aufwand sowohl bei der Implementierung der Validierungssoftware als auch bei der Schulung der Kontrolleure zu erwarten.

Zukünftig kann man von einer flächendeckenden Anbindung an das Internet ausgehen. Darauf aufbauend resultiert die Möglichkeit im UIC-918-3-konformen Barcode nur eine eindeutige Kennung unterzubringen und bei der Validierung die Ticketdaten über eine Datenverbindung zu einem Server anzufordern. Aktuell ist dies auf Basis der schnell fahrenden Züge und des schlechten Ausbaus der Funknetzwerke vor allem in den gebirgigen Bahnstrecken Österreichs nicht produktiv einsetzbar.

Index

Aztec 4, 9, 12, 14, 22, 23, 24, 25, 26, 27, 30, 36, 37, 41, 44, 47, 66, 75, I, IV, V, VI, VII, VIII, IX, XII

Barcode Library 41, 45, 46

Complex 55, 67, 68, 71, 73, 74, II, V, VIII, X, XIII

DEFLATE 27, 32, 53, 54, 69

HTML x, 41, 45

Huffman 27, 28, 29, 55, 56, 57, 74, 78

JSON 34, 35, 48, 50, 67, 68, 70, 78, II, V, VIII, X, XIII

PreCompressed 53, 67, 68, 69

RCT2 x, 4, 5, 6, 7, 8, 12, 17, 38, 42

SimpleText 52, 67, 68, 69, 71, 72, 74

UIC-918-3 9, 12, 13, 15, 16, 17, 19, 21, 25, 30, 36, 37, 38, 39, 44, 45, 47, 69, 73, 74, 75

XML x, 33, 41, 44, 48, 49, 66, 67, 68, 70, II, V, VIII, X, XIII

YAML x, 35, 48, 51, 67, 68, 70, 72, 78, III, VI, VIII, XI, XIV

Literatur

- /UIC918-2/ UIC: UIC CODE 918-2 3rd edition, Paris, UIC, 2008, ISBN 2-7461-1510-1
- /UIC918-3/ UIC: UIC CODE 918-3 1st edition, Paris, UIC, 2007, ISBN 2-7461-1307-4
- /AnKr2012/ Ertel, Wolfgang: Angewandte Kryptographie, München, Hanser, 2012, ISBN 978-3-446-42756-3
- /HBdal2002/ Lenk, Bernhard: Handbuch der automatischen Identifikation Band 2, München, Verlag Lenk Monika, 2002, ISBN 3935551010
- /InDaCo2005/ Khalid, Sayood: Introduction to Datacompression, USA, Morgan Kaufmann, 2005, ISBN 012620862X
- /HBaC1996/ Alfred Menezes, Paul van Oorschot, Scott Vanstone: Handbook of Applied Cryptography, USA, CRC Press, 1996, ISBN 0-8493-8523-7
- /Javal2011/ Ullenboom, Christian: Java ist auch eine Insel, Bonn, Galileo Press, 2011, ISBN 978-8362-1802-3
- /SWT2005/ Andreas Spillner, Tilo Linz: Basiswissen Softwaretest 3. Auflage, dpunkt.verlag GmbH, 2005, ISBN 3-89864-358-1

- /UIC918-3*/ DB: B@hnDirekt Interoperabilität Barcode DB Online-Ticket VDV KA,
http://www.bahn.de/p/view/mdb/bahnintern/verbund/barcode/mdb_97439_interoperabilitaet_barcode_db_online-ticket_vdv-ka_v1_3.pdf, verfügbar 29.07.2014, 11:48
- /UriASCII/ Wikipedia: American Standard für Information Interchange,
http://de.wikipedia.org/wiki/American_Standard_Code_for_Information_Interchange, verfügbar 29.07.2014, 12:01
- /UriBase64/ Brunner, Arndt: Der Kodierungsstandard Base64,
<http://www.arndt-bruenner.de/mathe/scripts/base64.htm>, verfügbar 29.07.2014, 12:05
- /UriJSON/ json.org: Einführung in JSON, <http://json.org/json-de.html>, verfügbar 29.07.2014, 12:20
- /UriReedS/ Martin Riley, Iain Richardson: Reed-Solomon Codes,
http://www.cs.cmu.edu/~guyb/realworld/reedsolomon/reed_solomon_codes.html, verfügbar 29.07.2014, 12:31
- /RedeFischer/ Dr. Heinz Fischer: Rede zum Nationalfeiertag 26.10.2012,
<http://www.bundespraesident.at/newsdetail/artikel/nationalfeiertag-die-rede-des-bundespraesidenten-im-wortlaut/>, verfügbar 05.08.2014
- /HTRosetta/ Rosettacode.org: Implementierung Huffman-Code,
http://rosettacode.org/wiki/Huffman_coding#Java, verfügbar 05.08.2014
- /UriYAML/ Yaml.org: YAML Specification 1.2,
<http://www.yaml.org/spec/1.2/spec.html> verfügbar 06.08.2014
- /UriUIC/ uic.org: UIC Vorstellung, <http://www.uic.org/spip.php?article529> verfügbar 06.08.2014

Anlagen

Testergebnisse „Minimal“	I
Testergebnisse „Maximal“	IV
Testergebnisse „MaximalSpec“	VII
Testergebnisse „BestehendeJK“	IX
Testergebnisse „DifferentData“	XII

Anlagen, Testergebnisse „Minimal“

Ticket

CIV				Fahrschein				A Personen 01		B		
28.07.2014 00:01 bis 29.07.2014 00:01												
Datum	Zeit	Von				-> Nach				Datum	Zeit	Klass.
		Platt				Velm						
										001,00.-		

Simple Text (Hersteller spezifische Daten)

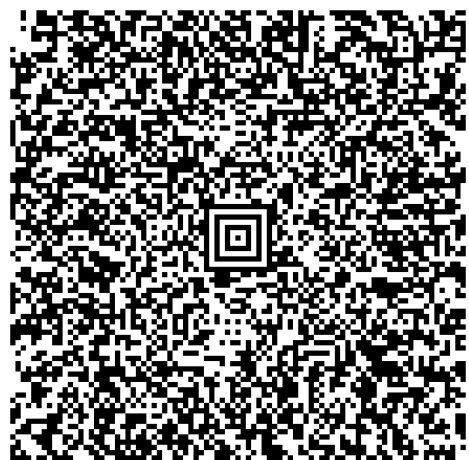
\\E\\1\\1\\M\\01011990\\A B\\FS\\1\\28.07.2014
00:01\\29.07.2014
00:01\\001\\00725\\00511\\001B01

Aztec-Barcode



Precompressed

eJyLcY0xBELfGANDA0NDS0uDGEcFpxi3YKC
YkYWegbmekYGhiYKBgZUBUMASTcA-
AjM2NTIGkqaFhTAxIxMnAEABkzROQ



Complex

I7Z7Y6YRZANDY2e2CmSAAIpIAACIPtgFqgP+
AAAAAJT6hPsMAw==



XML

```
<IN-  
DA><WKZ>E</WKZ><KID>1</KID><PID>1</PI  
D><GS>M</GS><GEB>01011990</GEB><NA  
ME>A  
B</NAME><IDT>FS</IDT><IDI>1</IDI><VFR>  
28.07.2014 00:01</VFR><VTO>29.07.2014  
00:01</VTO><STR>001</STR><ZONE><ZAN>  
00725</ZAN><ZEN>00511</ZEN><ZV1/><ZV2  
/></ZONE><ART><ARTI>001B01</ARTI></AR  
T></INDA>
```



JSON

```
{"GEB":"01011990","STR":"001","NAME":"A  
B","IDI":"1","PID":"1","ZAN":"00725","ZEN":"005  
11","VTO":"29.07.2014  
00:01","WKZ":"E","GS":"M","ART":["001B01"],"I  
DT":"FS","KID":"1","VFR":"28.07.2014  
00:01","ZV2":"","ZV1":""}
```



YAML

```
{GEB: '01011990', STR: '001', NAME: A B, IDI:  
'1', PID: '1', ZAN: '00725', ZEN: '00511', VTO:  
'29.07.2014 00:01', GS: M, ART: [001B01], IDT:  
FS, WZ: E, KID: '1', VFR: '28.07.2014 00:01',  
ZV2: "", ZV1: ""}
```



Anlagen, Testergebnisse „Maximal“

Ticket

Fahrerschein				Antoinella Mustermann Personen 09	
CV					
28.07.2014 13:42 bis 29.12.2014 20:59					
Datum	Zeit	Von	-> Nach	Datum	Zeit
		Hetzmansdorf-Wuller	Enzersdf. bei Staatz		
Über Leobendorf Über Langenzerdorf Über Floridsdorf					
Im Preis enthalten sind 20% Mehrwertsteuer				123.89.-	

Simple Text (Hersteller spezifische Daten)

\E\123456789012345678\0987654321\W\31021
987\Antoinella Muster-
mann\FS\ABCDEFHIJK\28.07.2014
13:42\29.12.2014
20:59\001\00635\00715\00425\00220\001B011
02S24346S35978S68

Aztec-Barcode



Precompressed

eJw1jTsOwkAMRK+SE0T22PtLI0AiPqLagsZNi
hRI-
sEgQ7s9GiGKeRm+KsdEYos6HmOjFjFIM3qmA
7WrCBK7C+rl+b2W53+fm8nmvy+sxI2JTtn7Y7c
fpcDydDbGI0IJYG5ZOYUgt4ydAnUtGxDVeXG
XgjYqNAG3bQFz/MITUZ3EpxOzjFwZ1KU4=

Aztec-Barcode konnte nicht erzeugt werden.

Complex

I75lhtxzfqn3zLDbjm+xOt/PG2GXnthp/72s/kpUw
58d5LdP/+x8XWxsmjSkm9sP9YKZQGbCrlCda
U+2AT2BZYDUgG4lPqE++p8j5hlhxHLbXm+PF
2wB



XML

```
<IN-  
DA><WKZ>E</WKZ><KID>123456789012345678</KID>  
<PID>0987654321</PID><GS>W</GS><GEB>31021987  
</GEB><NAME>Antoinella Muster-  
mann</NAME><IDT>FS</IDT><IDI>ABCDEFGHIJK</IDI><  
VFR>28.07.2014 13:42</VFR><VTO>29.12.2014  
20:59</VTO><STR>001</STR><ZONE><ZAN>00635</Z  
AN><ZEN>00715</ZEN><ZV1>00425</ZV1><ZV2>0022  
0</ZV2></ZONE><ART><ARTI>001B01</ARTI><ARTI>1  
02S24</ARTI><ARTI>346S35</ARTI><ARTI>978S68</A  
RTI></ART></INDA>
```

Aztec-Barcode konnte nicht erzeugt werden.

JSON

```
{"GEB":"31021987","STR":"001","NAME":"Antoi  
nella Muster-  
mann","IDI":"ABCDEFGHIJK","PID":"0987654321  
","ZAN":"00635","ZEN":"00715","VTO":"29.12.2  
014  
20:59","WKZ":"E","GS":"W","ART":["001B01","1  
02S24","346S35","978S68"],"IDT":"FS","KID":"1  
23456789012345678","VFR":"28.07.2014  
13:42","ZV2":"00220","ZV1":"00425"}
```

Aztec-Barcode konnte nicht erzeugt werden.

YAML

```
{GEB: '31021987', STR: '001', NAME: Antoinella  
Mustermann, IDI: ABCDEFHIJK, PID:  
'0987654321', ZAN: '00635', ZEN: '00715', VTO:  
'29.12.2014 20:59', GS: W, ART: [001B01,  
102S24, 346S35, 978S68], IDT: FS, WZ: E,  
KID: '123456789012345678', VFR: '28.07.2014  
13:42', ZV2: '00220', ZV1: '00425'}
```

Aztec-Barcode konnte nicht erzeugt
werden.

Anlagen, Testergebnisse „MaximalSpec“

Ticket

CIW		Fahrschein		Antoinella Mustermann Personen 09	
28.07.2014 13:42 bis 29.12.2014 20:59					
Datum	Zeit	Von -> Nach		Datum	Zeit
		Hetzmannsdorf-Wuller Enzersdf. bei Staatz			
Klass.					
In unserem Land, das heute seinen Nationalfeiertag begeht, leben mehr als acht Millionen Menschen in neun Bundesländern u					
				123.89.-	

Simple Text (Hersteller spezifische Daten)

\E\123456789012345678\0987654321\W\31021
 987\Antoinella Muster-
 mann1435\FS\ABCDEFHIJK\28.07.2014
 13:42\29.12.2014
 20:59\001\00635\00715\00425\00220\001B011
 02S24346S35978S68

Aztec-Barcode

Aztec-Barcode konnte nicht erzeugt werden.

Precompressed

eJw1jTsOwkAQQ6+SE0QzntlfugQS8RHVFjTTP
 EiBBI-
 sE4f5shChsWc+ybKMxRJ0PMdE/GaUYvFMB2
 9WECVyB9WV93spyv8/N5fNel9djLoVVnE3Z+
 mG3H6fD8XQ2xJZCC2JtWDqFIbWMHwB1Lhk
 RV/m6JAq8uWJzgLZuIK6fGSrqs7gUYvbx6M
 dKhs=

Aztec-Barcode konnte nicht erzeugt werden.

Complex	
I75lhtxzfqn3zLDbjm+xOt/PG2GXnthp/72s/kpUw 58d5LdP//3DLax8XWxsmjSkm9sP9YKZQGbCrI CdaU+2AT2BZYDUgG4IPqE++p8j5hlhXHLbXm +PF2wB	Aztec-Barcode konnte nicht erzeugt werden.
XML	
<IN- DA><WKZ>E</WKZ><KID>123456789012345678</KID> <PID>0987654321</PID><GS>W</GS><GEB>31021987 </GEB><NAME>Antoinella Muster- mann1435</NAME><IDT>FS</IDT><IDI>ABCDEFHIJK</I DI><VFR>28.07.2014 13:42</VFR><VTO>29.12.2014 20:59</VTO><STR>001</STR><ZONE><ZAN>00635</Z AN><ZEN>00715</ZEN><ZV1>00425</ZV1><ZV2>0022 0</ZV2></ZONE><ART><ARTI>001B01</ARTI><ARTI>1 02S24</ARTI><ARTI>346S35</ARTI><ARTI>978S68</A RTI></ART></INDA>	Aztec-Barcode konnte nicht erzeugt werden.
JSON	
{"GEB":"31021987","STR":"001","NAME":"Antoi nella Muster- mann","IDI":"ABCDEFHIJK","PID":"0987654321 ","ZAN":"00635","ZEN":"00715","VTO":"29.12.2 014 20:59","WKZ":"E","GS":"W","ART":["001B01","1 02S24","346S35","978S68"],"IDT":"FS","KID":"1 23456789012345678","VFR":"28.07.2014 13:42","ZV2":"00220","ZV1":"00425"}	Aztec-Barcode konnte nicht erzeugt werden.
YAML	
{GEB: '31021987', STR: '001', NAME: Antoinella Mustermann1435, IDI: ABCDEFHIJK, PID: '0987654321', ZAN: '00635', ZEN: '00715', VTO: '29.12.2014 20:59', GS: W, ART: [001B01, 102S24, 346S35, 978S68], IDT: FS, WZ: E, KID: '123456789012345678', VFR: '28.07.2014 13:42', ZV2: '00220', ZV1: '00425'}	Aztec-Barcode konnte nicht erzeugt werden.

Anlagen, Testergebnisse „BestehendeJK“

Ticket

CV				Fahrschein				Czanitz Personen 01		Gregor		
01.12.2013 00:01 bis 31.01.2015 23:59												
Datum	Zeit	Von				-> Nach				Datum	Zeit	Klass.
		Paasdorf				Wie						
Über Auersthal												
										345.10.-		

Simple Text (Hersteller spezifische Daten)

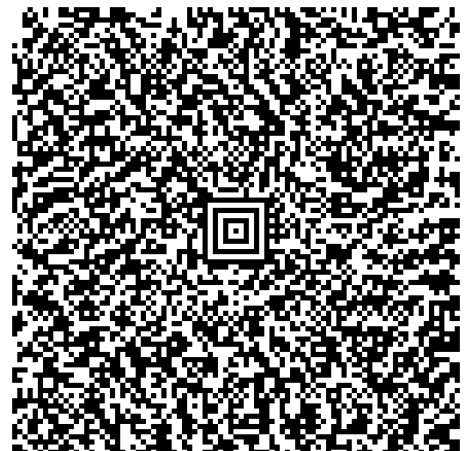
\E\745-344-12\01055\M\17031965\Gregor Cza-
 nitz\FS\0727863-BH\01.12.2013
 00:01\31.01.2015
 23:59\365\00515\00510\00310\0099B01

Aztec-Barcode



Precompressed

eJwlzDsOw-
 jAQhOGr+AK2ZrzeGKdMxKOhop2GAiEakCKq
 nJ6VaL7i12h0VG+erbXMKhDuuoodxjG5ztvj+dn
 Sut/fr++u003otR8my8sl1oW1VNASMIMyImgRP
 FWbfcjiAnD+RWihMMYC/gCO8x3v



Complex

IzhtWXLDCsvli-
fU7bWOguULVjTqf7N+rF0lbGxPPnN/GVZUd2
Ce2AAIrCgs/lxtYBAYD/AJsAACddUJ9gwE



XML

```
<INDA><WKZ>E</WKZ><KID>745-344-  
12</KID><PID>01055</PID><GS>M</GS><GE  
B>17031965</GEB><NAME>Gregor Cza-  
nitz</NAME><IDT>FS</IDT><IDI>0727863-  
BH</IDI><VFR>01.12.2013  
00:01</VFR><VTO>31.01.2015  
23:59</VTO><STR>365</STR><ZONE><ZAN>  
00515</ZAN><ZEN>00510</ZEN><ZV1>00310  
</ZV1><ZV2></ZONE><ART><ARTI>099B01  
</ARTI></ART></INDA>
```



JSON

```
{"GEB":"17031965","STR":"365","NAME":"Greg  
or Czanitz","IDI":"0727863-  
BH","PID":"01055","ZAN":"00515","ZEN":"00510  
","VTO":"31.01.2015  
23:59","WKZ":"E","GS":"M","ART":["099B01"],"I  
DT":"FS","KID":"745-344-12","VFR":"01.12.2013  
00:01","ZV2":"","ZV1":"00310"}
```



YAML

```
{GEB: '17031965', STR: '365', NAME: Gregor  
Czanitz, IDI: 0727863-BH, PID: '01055', ZAN:  
'00515', ZEN: '00510', VTO: '31.01.2015 23:59',  
GS: M, ART: [099B01], IDT: FS, WZ: E, KID:  
745-344-12, VFR: '01.12.2013 00:01', ZV2: "",  
ZV1: '00310'}
```



Anlagen, Testergebnisse „DifferentData“

Ticket

Fahrerschein				Czanitz Gregor Personen 01		
01.12.2013 00:01 bis 31.01.2015 23:59						
Datum	Zeit	Von	-> Nach	Datum	Zeit	Klass.
		Paasdorf	Wie			
Über Auersthal						
				345.10.-		

Simple Text (Hersteller spezifische Daten)

\E\745-344-12\01055\M\17031965\Roman Stephan Friedrich\FS\0727863-BH\14.05.1999
15:46\29.11.2014
20:37\365\00515\00510\00310\0099B01

Aztec-Barcode



Precompressed

eJwlzDEOw-
jAQRNGr+AK2Zrxem00ZiYiGhrTblliUFEAUcX9
hieb97vvZW9EopURmB6HqV2eD0Kr67fO6v8P
8Xfa1dzq25Xlsj9Wn2dFyO1WJ48VZEjTRzAJ1
KNWzJTJlsISMQZpLfwHKv+hK12E2gj8TwCFR



Complex

IzhtWXLDCsvli-
fU7bWOXa6f4jvInc/w1bJUWzTuxdCGxsTz5zfxl
WVHdn///4yB2QqjAmrlxtYBAYD/AJsAACddUJ9
sCA==



XML

```
<INDA><WKZ>E</WKZ><KID>745-344-  
12</KID><PID>01055</PID><GS>M</GS><GE  
B>17031965</GEB><NAME>Roman Stephan  
Fried-  
rich</NAME><IDT>FS</IDT><IDI>0727863-  
BH</IDI><VFR>14.05.1999  
15:46</VFR><VTO>29.11.2014  
20:37</VTO><STR>365</STR><ZONE><ZAN>  
00515</ZAN><ZEN>00510</ZEN><ZV1>00310  
</ZV1><ZV2></ZV2></ZONE><ART><ARTI>099B01  
</ARTI></ART></INDA>
```



JSON

```
{"GEB":"17031965","STR":"365","NAME":"Roma  
n Stephan Friedrich","IDI":"0727863-  
BH","PID":"01055","ZAN":"00515","ZEN":"00510  
","VTO":"29.11.2014  
20:37","WKZ":"E","GS":"M","ART":["099B01"],"I  
DT":"FS","KID":"745-344-12","VFR":"14.05.1999  
15:46","ZV2":"","ZV1":"00310"}
```



YAML

```
{GEB: '17031965', STR: '365', NAME: Roman  
Stephan Friedrich, IDI: 0727863-BH, PID:  
'01055', ZAN: '00515', ZEN: '00510', VTO:  
'29.11.2014 20:37', GS: M, ART: [099B01], IDT:  
FS, WZ: E, KID: 745-344-12, VFR: '14.05.1999  
15:46', ZV2: "", ZV1: '00310'}
```



Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Wien, den 24.11.2014

Ing. Roman Waitz